-CPE191 / EEE193B - Senior Design Project II

End of Project Report

QB : The Quarantine Buddy

Presented to the faculty of the Department of Electrical and Electronic Engineering
California State University, Sacramento

Team 7

Victor Lugo

Alex Rodriguez

Zach Tapia

Jayson Thorpe

Instructors:

Russ Tatro

Neal Levine

James Cottle, PhD

SPRING

MAY 3rd, 2021

TABLE OF CONTENTS

Executive Summary iii
Abstract 1
I. Introduction 1
II. Societal Problem 4
III. Design Idea 10
IV. Funding 16
V. Project Milestones 17
VI. Work Breakdown Structure 17
VII. Risk Assessment 23
VIII. Design Philosophy 26
IX. Deployable Prototype Status 26
X. Marketability Forecast 29
XI. Conclusion 32
References 34
Glossary 37
Appendix A. User Manual A-1
Appendix B. Hardware B-1
Appendix C. Software C-1
Appendix D. Mechanical Aspects D-1
Appendix E. Vendor Contacts E-1
Appendix F. Resumes F-1
Appendix G. Project Timeline & Work Breakdown Structure G-1

TABLE OF FIGURES

Figure 1. Stress Rates in Adults from May-July 2020 [1] **4**
Figure 2. Unemployment Rate Aug. 2000 – Aug. 2020 [3] **5**
Figure 3. Psychological Distress Symptoms within a Week of Quarantine [4] **6**
Figure 4. Parental Source of Stress in America [6] **6**
Figure 5. Negative Effects of Stress [9] **7**
Figure 6. Alpha-Stim [13] **8**
Figure 7. % of Patients Who Reported Improvement After Using Alpha-Stim [13] **9**
Figure 8. Flow Diagram of Proposed Design [30] **11**
Figure 9. Diagram of the Seebeck Effect [31] **13**
Figure 10. ZTP-115 Infrared Temperature Sensor [31] **13**
Figure 11. Flow Diagram of Amazon Web Services [30] **16**
Figure 12. Example of GSR Spike with components for deriving magnitude [32] **18**
Figure 13. Temp Sensor Amplification Circuitry [30] **19**
Figure 14. Voltages of GSR & Skin Temp reacting to introduction of a Stressor [30] **19**
Figure 15. Voltage as a function of time for Temperature Infrared Sensor [31] **19**
Figure 16. GSR amplification circuit [32] **20**
Figure 17. GSR during deep breathing [30] **27**
Figure 18. Voltage output of Skin Temp sensor [30] **27**

TABLE OF TABLES

Table 1. Bill of Materials **16**
Table 2. Cost Breakdown of a single QB unit **31**

ELEVATOR PITCH

Social distancing has led to an increase in stress levels and related symptoms. A robotic companion with the means to monitor and alleviate these symptoms while a user self-isolates is called for.

EXECUTIVE SUMMARY

The onset of the COVID-19 pandemic has led to worldwide increases in the levels of stress, anxiety, loneliness, and depression. Implementation of social distancing has led to the closure of schools and businesses causing increased isolation and job loss, both of which are major factors in exacerbating stress and anxiety. Given the widespread increase in stress and by extension the negative effects it has on physical health and wellbeing, a design to assist in detecting, relieving, and managing stress will be developed. This report details the development process and technical details of the Quarantine Buddy (QB), as an experimental solution to this rise in stress levels induced by the pandemic.

Firstly, this report presents the details of the problem of rising stress that is facing society. This portion of the report discusses statistics associated with the problem, the technical context, and issues one would have to address in order to address the problem itself. Already existing engineering solutions to similar issues are also discussed in this section.

The next section of this report introduces our team's proposed solution to the presented problem. This section introduces the general concept of QB and the ways it will address the issues laid out in the preceding section. QB's functionality is then laid out, followed by a comparison to some of the pre-existing products discussed in the previous section as well as some details showing how QB is unique and different to some of these other approaches. The report then goes on to discuss each of QB's features in detail.

The remainder of this report concerns itself with the development process of QB. The fourth section lays out in detail the steps of this development that are considered to milestones towards QB's completion. This section also details the smaller achievements that must successfully take place for these milestones to be fully implemented.

We then briefly discuss the funding behind the QB project, in addition to laying out the price breakdown of all the components that made it into the final prototype.

Next is the discussion of the work breakdown structure, which serves to organize the work needed to be completed and divides it among the team according to our ability and interest. The work breakdown also serves as a template for our timeline as to when we foresee certain work being completed. This acts as a sort of schedule we attempted to hold ourselves to for the remainder of the project.

After the work breakdown structure follows our risk assessment of the whole project. This includes a projection of everything we could think of (within reason) that could go wrong during the development phase, including a risk assessment matrix which seeks to assign some of the more catastrophic predicted issues a weight based on the likelihood of it occurring and the severity to which its occurrence would impact the entire project. The very event that influenced the creation of QB, the COVID-19 pandemic, appropriately enough presented its fair share of challenges to be overcome throughout the entire process.

The following section is the current status of the QB prototype and is updated every iteration of this report. In this final version of this report this section will detail the technical specifications and status of QB and all the features that comprise the whole of its functionality.

The final section of this report before the conclusion explores the current state of the market for QB and other products that would serve as its competitors. We compare and contrast different aspects of these including price, feature set, practicality and so on. This served as a sort of insight on the practicality of QB itself and served as a fascinating topic of discussion and conjecture within the team and with our overseeing professor.

ABSTRACT

As a result of increased social distancing due to the ongoing COVID-19 pandemic, many find themselves increasingly isolated from friends, family, and other means of support which they might otherwise have leaned upon as a source of relief. With so many people today living in increasingly stressful conditions, the opportunity for worsening mental health increases as well. Knowing that these in turn have a negative impact on physical health conditions like high blood pressure, obesity, and heart disease to name a few, the need to mitigate stress is becoming increasingly evident. We propose a solution; Project QB. QB, short for Quarantine Buddy is a robotic companion designed especially for those missing out on their normal support net because of social distancing and self-isolation during the pandemic. QB's functionality can be divided up into a few main features. Biometric monitoring, Bluetooth Communication, Robotic Movement, and a Voice User Interface. These features were divided up among the team and will gradually converge to the end result of the project. The nature of a Senior Design under social distancing guidelines has necessitated this modular design strategy. Biometric readings are used to detect when the user is stressed, and theses readings are wirelessly transmitted to QB, who will then assist in relieving the user's stress. The User Voice Interface not only controls the flow of QB's behavior but, when combined with the Robotic Movement feature serves to provide QB with the semblance of a personality. QB strives to fill the social support void left by the increase in social distancing and self-isolation during the ongoing COVID-19 pandemic.

KEYWORD INDEX – Biometrics, Companion Coronavirus, COVID-19, Isolation, Mental health, Pandemic, Quarantine, Robotics, Social distancing, Stress, Vital signs

I. INTRODUCTION

The team first researched a societal problem that had an impact on a large population to which we could provide a technological solution. A large population was determined to be over 1 million people. The technical solution needed to implement the skills and techniques expected of Senior level undergraduate electrical and computer engineers. Issues related to the COVID 19 pandemic were of primary interest, specifically stress.

Stress related to the spread of the virus and the resulting social distancing caused an increase in symptoms of mental health issues. The increased stress and related symptoms were further researched and shown to have detrimental effects on overall health and wellbeing. Exacerbating these issues were limited access to mental health support. An engineering solution to help mitigate the increase in stress was to design an interactive robot to monitor and support user stress levels. This "social robot" incorporates a wearable stress monitor that communicates with the robot itself to provide stress relief. Communication is provided through the Bluetooth Serial Port Protocol (Bluetooth SPP). Stress relief is accomplished through interaction with media controlled by Amazon Web/Voice Services as well as physical interaction with the robot. The voice user interface and the robotic gestures of the robot seek to simulate a personality to further maintain a sense of personality and friendliness in the virtual companion. The robot has been dubbed QB, short for Quarantine Buddy.

Our social robot's design incorporates a wearable stress detection monitor. This monitor is comprised of an array of sensors for galvanic skin response, infrared skin temperature, and heart rate. The monitor contains a small wearable microcontroller device to process the sensor signal input to constantly calculate the user's stress score. When the stress score is found to be above a given threshold value, a moment of stress is triggered which is then sent to the main body of QB via an integrated Bluetooth module in the wearable device.

The sensor array contains a sensor that detects the resistance of the skin and outputs a voltage. The inverse value of this voltage is then the conductance. The GSR uses amplification circuitry based around the LM324, however, this was available in a pre-built, off the shelf module. The LM324 op-amps are configured into an instrumentation op-amp, which is good for limiting noise amplification. Sweat on the fingers would lower the potential difference and result in a lower reading (low resistance /higher conductivity). Dry fingers would result in a higher reading (higher resistance/lower conductivity).

The sensor array also contains an infrared sensor to detect changes in skin temperature (ST). A ZTP-115 was used for this. This sensor is a thermophile which employs the Seebeck effect. This allows the sensor to passively output a voltage based on the infrared light that is absorbed. Amplification circuitry using the LM324 op-amp boosts the magnitude of this voltage signal to values differentiable by the microcontroller's ADC. The op-amp was to be powered by a small watch battery outputting 3V. The amplifier was designed to have a gain of 1200 V/V. Because the amplifier output to an ADC was capable of reading only positive values with reference to GND, the inputs were biased to 1.5V.

The microcontroller selected for the wearable stress monitor was the Arduino Lilypad. This device uses the 8-bit ATMega328p. The microcontroller uses 10-bit ADCs to convert the input from the sensor circuits into digital values. Initially, a +5V supply was available, which required at least 4.9 mV of differentiation from the amplifier circuits. Conversion from the original Arduino Uno development kit to the Arduino Lilypad meant lowering the supply to +3V. This increases the accuracy of the sensor output, since (2^10)/3V > (2^10)/5V, allowing for increased precision of voltage values by the ADC. Coding of the stress detection algorithm, IoT connectivity and robotic motor control was carried out in the Arduino IDE. A GitHub repository was used to handle version control.

An algorithm using these biometric signals (GSR and ST) was researched and programmed into the microcontroller. The algorithm calculates the magnitude of a user's response to specific stressors in time. Characteristics of the measured values over time are then used to determine five components of an overall stress score. Each component of the final stress score were related to one of five characteristics related to GSR or ST. These five characteristics were: 1. GSR increase. 2. Skin Temperature decrease time. 3. GSR rise time. 4. GSR slope. 5. Multiple onsets within 10 seconds. An onset is defined as a point in time when the GSR begins to rise and continues to rise for between 2 and 5 seconds before peaking. A weighting system was used to determine the magnitude of each of the five stress score components. Depending on how much the measured values satisfied the algorithm a score of 0, 1, or 2 was given to each component. Each component was then multiplied to a predetermined component weight. These were set at 30, 25, 20, 15, and 10 for rule 1, 2, 3, 4 and 5 respectively. The maximum score was 200 while the minimum score was 0. A threshold score was defined as 175. Anything above this value indicated a moment of stress was detected.

The algorithm senses phasic stress, which results from a specific stress trigger at a specific point in time, as opposed to tonic stress, which is a response to background stress triggers that do not directly correlate to a stimulus at a specific time. Tonic stress levels are important to detect and manage, however detection of this type of stress was deemed much more difficult. The team decided to focus on phasic stress detection, which seemed more easily detectable using the sensory equipment that was readily available.

Wireless communication between the stress detection monitor and the robot was deemed necessary. Many devices available before and made available during the pandemic that are marketed as stress relief/monitoring products incorporate this feature. This module leverages off-the-shelf Bluetooth modules that use the Serial Port Protocol. This protocol keeps programming of communication between the sensor array microcontroller and the main robotic system relatively simple. Off the shelf HC-05 modules are designed to operate seamlessly with Arduino microcontrollers. A pair of these modules was programmed using AT commands and the Arduino IDE serial monitor. The master was connected to the stress monitor and the slave was connected to the robotic module. The BAUD rate chosen was 115200 to comply with protocol needed for internet connectivity in other feature sets.

In order to facilitate a livelier companion, part of our design includes QB performing physical gestures while interacting with the user. Our robot design would thus require a body two arms and a head. In total a total of seven servo motors were used to give QB varying gestures. Each arm had three motors to perform actions such as wave, arm raise, and fist pump. Last, there was one in the neck to perform a simple had nod. The actions performed were based on phrases spoken by the user which were programmed in AWS as spoken earlier. The robot also had LED's that served as lights. They would light up when stress is detected and while the user was interacting with QB. The robot triggers had to be controlled an ESP32, AWS IoT and Lambda. The ESP 32 used Bluetooth to communicate with the wearable and WIFI to communicate with AWS. The AWS IoT had a shadow object which reflected the state of QB to be used by Lambda and any changes provided by Lambda. QB required many components to be synchronized for timely responses to the user.

Research of the societal problem and development and testing of the prototype required two semesters. A timeline was developed and modified over the course of the near year-long process. This timeline helped breakdown the specific feature sets required for prototype delivery as well as the other tasks required for project completion.

The initial societal problem was researched over the course of the summer (2020). Each team member was tasked with researching a specific problem of interest resulting from pandemic. Ideas developed revolved around stress management, mitigation of fires, home irrigation and exercise routine management. Stress increase and stress management were chosen as the team's societal problem to focus on.

After laying out the feature set of QB and establishing the design idea contract, a project timeline and work breakdown structure (WBS) was developed. QB is a relatively large project and includes features employing techniques in computer programming, analog circuit design, and robotics. The prototype's engineers needed time to research individual feature-set implementation methods and in some cases develop entirely new skill sets to deal with the problems of development. Thus, a timeline detailing each facet of the project was essential in determining which aspects of the project to delve deeply into and which ones to scale down, change or eliminate. Most of the originally proposed features were fully implemented.

Following selection of the stress problem, ideas were laid out to develop an engineering solution: the social robot named QB, the Quarantine Buddy. The feature-set of QB discussed previously and detailed in Section III was then developed over the next few months. The feature-set is unique compared to the approach of many stress related products available in the market as of May 2021. This reflects the uniqueness of the circumstances in which it was developed. The prototype monitors stress and assists in its relief by combining a wearable monitor with a physical companionship robot with IoT capability.

The many features of QB warranted a thorough project risk assessment. During the early development stages, greater understanding of the difficulties of each feature's implementation was gained. This added awareness to each team member of the risks posed to QB's completion. After completion of the project timeline, a detailed risk-assessment was developed.

QB falls into the markets of stress relief/stress management and social robotics. A brief marketing report was developed to assess the feasibility of scaling production of QB within these markets. It was determined that these markets are growing and demand for products such as QB is increasing. The COVID 19 pandemic itself is a large growth driver, but so is changing perception and attitude toward mental health. Government stimulus packages were seen as another important growth driver.

Trying to address a problem related to the pandemic also requires adjusting a work schedule to be safe in the current pandemic induced lockdown environment. In order to work in an optimal manner and in industry standard, tasks will need to be worked on in parallel and all necessary components need to be known. Task without dependencies can be worked on in parallel and within this project implementing sensors, motor movements, Bluetooth communication, and Alexa communication. These tasks done separately allow the project to move at a faster pace. Completing these tasks move into the integration phase where syncing and further testing take place.

This project aims to have a voice interactive robot that monitors the user via wireless wearable with sensors. We know milestones include syncing motor movements with Alexa decisions and sending sensor to robot module via Bluetooth. The next major milestone is syncing Alexa decisions with sensor data. The rest of the project includes finetuning and improving QB interactions and decisions.

In the end we aim to provide companionship with stress detection, voice interaction, supportive gestures, and stress reducing activities. The timeline also includes assignments for the class which can impact the rate of project completion. However, using this timeline and adhering to dependencies makes managing this project more convenient and allows us to adjust to problems easier. Managing the project also requires looking into the future and determining possible risks that can slow or halt progress. Preparing for such includes determining potential impacts on progress and implementing some sort of mitigation strategy. Mitigation and risks revolve around the project's critical paths. Risks to the project's critical paths include the present and obvious social distancing. Impacts include working separately and higher chance of miscommunication. There also lies the possibility of not being able to successfully implement a component also may hinder the project. However, having mitigation strategies and foreseeing risks impacts allows us to have minimal delays to our timeline. This report outlines our strategies for the design process and implementing it using the most optimal methods.

II. SOCIETAL PROBLEM

Additional fear, anxiety and stress has increased across the globe as a result of coping with a new disease. A Kaiser Family Foundation tracking poll cites that in July 2020, 53% of adults in the United States reported that their mental health was been negatively impacted due to worry and stress over the coronavirus [1]. Specific negative impacts reported were difficulty sleeping, eating, increases in alcohol and substance consumption, and worsening chronic conditions. The poll shows that more than one in three adults in the U.S. have reported symptoms of anxiety or depressive disorder during the pandemic. Rising rates of anxiety disorder and depressive disorder have been observed during the COVID-19 pandemic. As seen in the chart of Figure 1, a steady increase in both was observed between May and July of 2020.

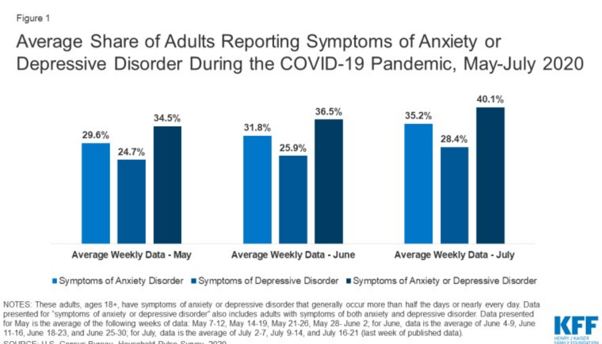


Figure 1. Stress Rates in Adults from May-July 2020 [1].

A. Added Stress from Social Isolation

Curbing the spread of COVID-19 has meant enforcement of social distancing. This has forcibly closed many businesses and had a heavy impact on the global economy. Figure 2 shows a spike in the unemployment rate of nearly 15% in the United States. Research from the National Center for Biotechnology Information (NCBI) states that “after unemployment, symptoms of somatization, depression, and anxiety were significantly greater in the unemployed than employed” [3]. Hence, the results of such widespread unemployment will take a heavy toll on the health of the general population.

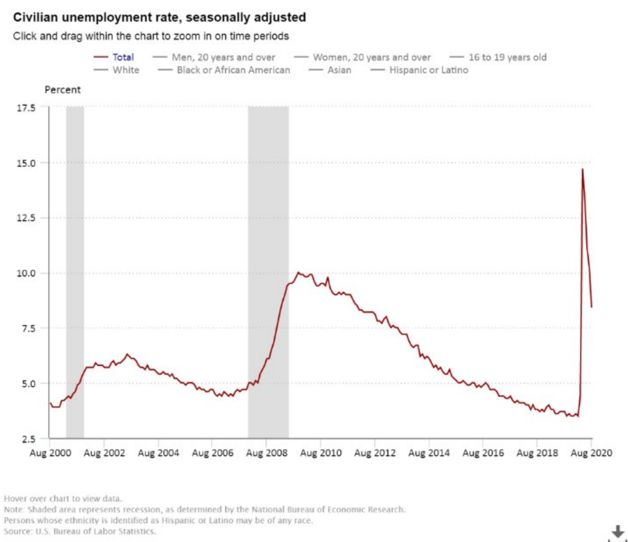


Figure 2. Unemployment Rate Aug. 2000 - Aug 2020 [3]

The convention of socializing in everyday life has been reduced abruptly for many people around the world due to corona. Isolating and social distancing in the pandemic has brought mental problems alongside the more obvious physical ones. Financial stability, domestic issues, and more are alleviated less from personal interactions due to the circumstances. Not only are the mentally vulnerable experiencing mental issues in this climate, but also those who were not in a vulnerable state. Shortly after the pandemic started in the United States, a study was done on the mental being of some citizens.

In mid-March, over 11,000 random people answered a survey to examine their well-being amidst the early pandemic. Figure 3 shows the results of this study, restricted to people who did not have a history of mental health conditions [4]. general population.

After analyzing the data, they found that one in four of this sample were experiencing psychological distress. And of that group 39% had symptoms of anxiety and depression. The result showed that psychological distress had increased since the start of the pandemic [4]. Addressing issues of mental health is always important and finding solutions can help being prepared for disasters such as this.

B. The Burden of School Closures

UNESCO cites “approximately 1.277 billion learners [...] currently affected due to school closures in response to the pandemic” [5]. 192 countries had closed all schools and universities, affecting more than 90 percent of the world’s learners”. Thus, one sixth of the world's population will suffer from increased isolation. In addition, with schools closed, parents must take on an added role of facilitating their children’s education, yet another source of added stress.

As seen from Figure 4, while adults were likely to report feeling stressed during pandemic, parents were especially prone to increased levels of stress. The added self-isolation, disruption of routines and missing out on major milestones were all shown to be significant stressors.

C. Increase in Suicide Rates

Another demonstration in the effects of pandemic on stress and related symptoms is in the increase in suicides. A survey by the CDC found that 11% of respondents reported suicidal ideation within the previous 30 days, as compared to 2018 when only 4% of respondents reported those thoughts [7]. At increased risk are marginalized groups such as low-income populations, the LGBTQ community and the homeless. Suicide prevention experts have noted that the pandemic will “exacerbate concerns [over these communities], as there are often more barriers to treatment and resources for these groups [8].”

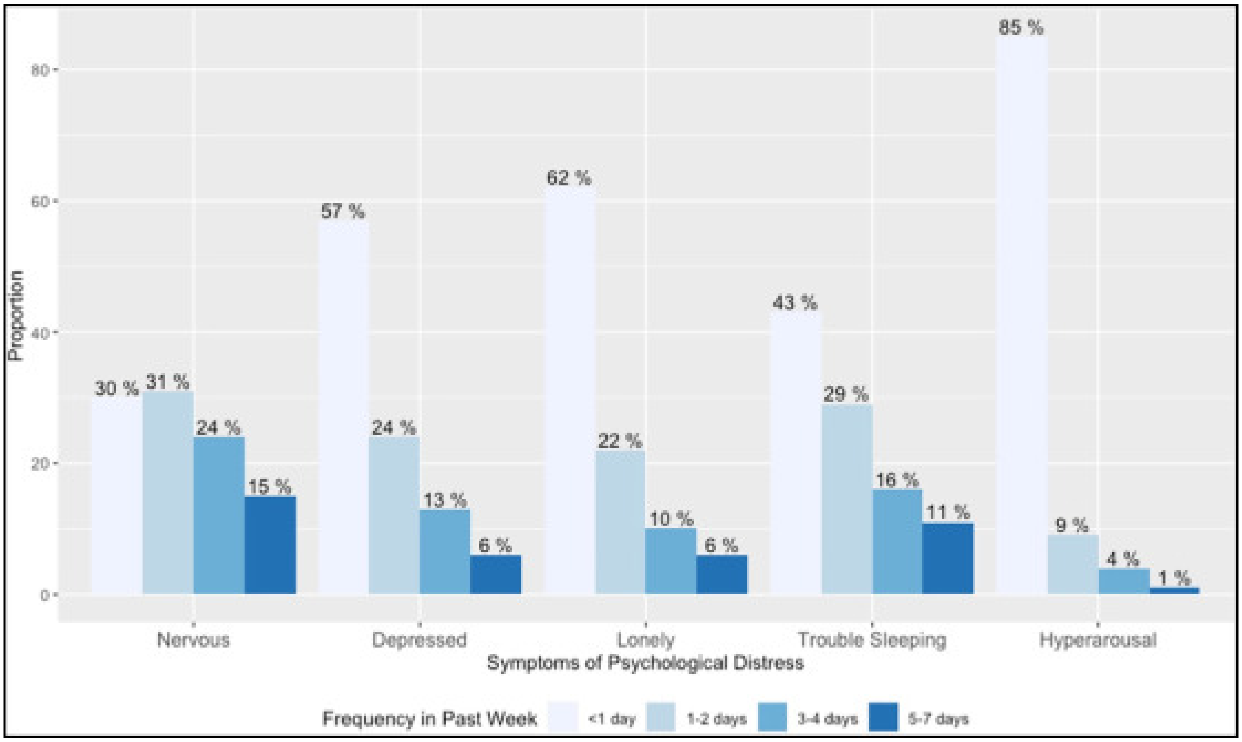


Figure 3. Psychological Distress Symptoms within a Week of Quarantine [4]

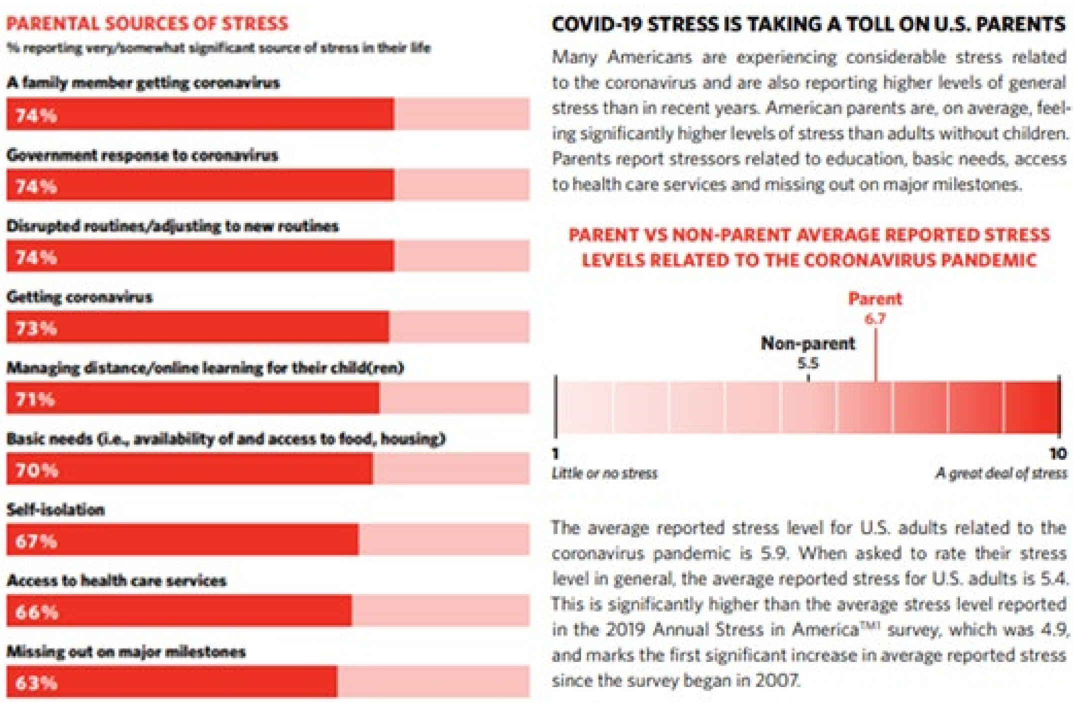


Figure 4. Parental Source of Stress in America [6]

D. Negative Effects of Stress

Stress can cause a multitude of negative effects in the body and the mind. It is important to understand that stress can have everlasting impacts on the body that further develop into serious deceases. See Figure 5 illustrates a variety of effects stress has across the entire human body. The

American Institute of Stress has verified that high or increased levels of stress can lead to episodes of depression and insomnia [9]. This means that stress is a gateway to other dangerous symptoms that may not be present for the individual and therefore increased the likelihood of chronic depression or substance abuse in some cases. Additionally, stress is the body's defensive mechanism when dealing with unwanted moments of fears. The body reacts by pumping blood and oxygen levels higher than usual and this causes an increment in blood pressure and heartrate.

This is potentially risky for many individuals as high blood pressure and heart rate could increase the risk of heart attacks. During a pandemic, such as that caused by COVID-19, one must be taking the right precautions to not become infected by the virus. The government has put into place many of the guidelines and regulations that can help flatten the curve of new COVID-19 cases. Many, however, still are forced to go to areas in which they may or may not be exposed to the virus, and this can lead to higher levels of stress. Longer periods of stress in the body can cause your immune system to weaken and therefore it may increase the risk to become infected by a virus such as COVID-19.

E. Proven Techniques of Alleviating Stress

Living a "stress-free" life would be incredible, however, not realistic. In the real world, everyone is subject to some form of stress one way or another. As described in earlier states of this report, many are suffering from a specific type of stress known as chronic stress. This type of stress is the natural reaction when worrying about what can happen to your loved ones, to your daily life, and your job security [9]. During a worldwide pandemic, stress and its negative effects on the human body and brain may seem inevitable but with early detection and proven techniques one can help mitigate and reduce stress. One of the ways engineers have been able to measure early stages of stress is using wearable devices and sensors.

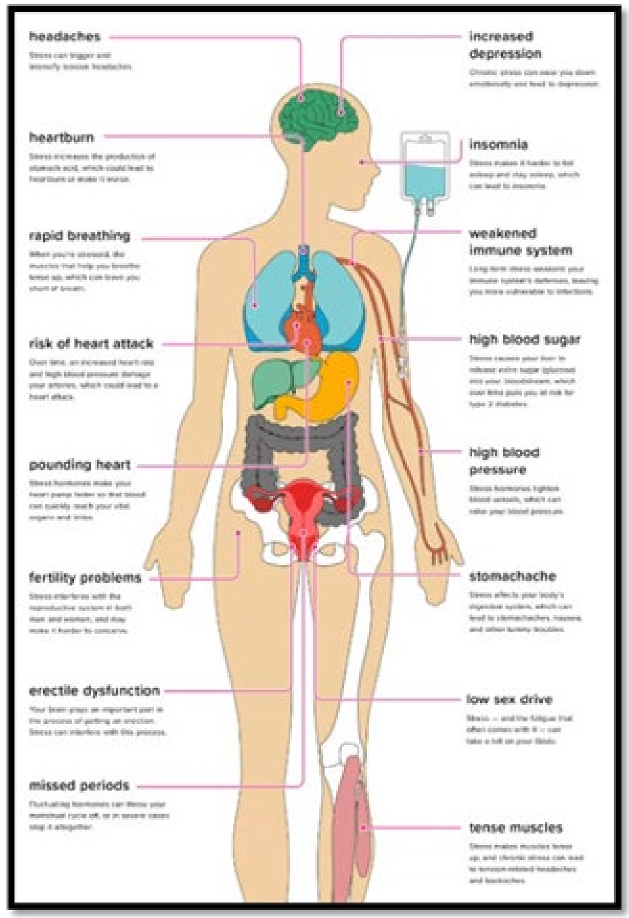


Figure 5. Negative Effects of Stress [9].

In a journal entry at the U.S. National Institutes of Health's National Library of Medicine Results, they show that wearable sensors helped detect MOS (Moments of Stress) with an 84% accuracy [10]. This technology proved to be successful in measuring stress in conjunction to other methods such as questionnaires and written diary entries stress events. The research obtained several data points that were vital in researching and understanding moments of stress. Galvanic skin response, cardiac activity, sweat gland activity, and skin temperature are all valid ways in which one can measure levels of stress. Engineers on this research focused on two major data points and those were: galvanic skin response and heart rate.

"There are parameters that are used to study the HRV (Heart Rate Variability) in the time domain such as mean value, the standard deviation of RR intervals, root mean square, etc. In the frequency domain, the most widely used method is low frequency (LF), high frequency (HF), LF/HF ratio, etc. [10]."

Other research has claimed to obtain an accuracy of 92.85% for the detection second- or third stress class using an EEG Headband [11]. An EEG or an electroencephalography device, is a sensor that can be used to monitor brain activity. Another approach that computer engineers and electrical engineers have used to measure levels of stress is using the power of machine learning algorithms and artificial intelligence.

In a study conducted in Boğaziçi University, stress recognition was improved by using machine learning (ML) algorithms such as Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), k Nearest Neighbors (kNN) and Fuzzy Logic classifiers [12].

Other engineering ways of combating signs of stress, depression and/or anxiety come in small packages such as the electrotherapy device ALPHA- STIM (shown in *Figure 6*). This device is a drug free solution treating patients that suffer from stress related pains such as anxiety, depression or insomnia [13]. *Figure 7* shows the percentage of users that noted decreases in their respective symptoms after using Alpha Stim for only 3 weeks.

The CDC recommends many ways of dealing with stress including taking deep breaths, stretching and meditation as well as getting plenty of sleep. It also recommends that, while social distancing is in place, connecting online through social media or by phone is helpful in relieving stress [7]. As regards to breathing exercise, one study on diaphragmatic breathing exercises showed the potential to reduce the consequences of stress in healthy adults both subjectively and physiologically [14].

Figure 6. Alpha-Stim [13].

Certain types of online media interaction have also proven beneficial in the reduction of stress. A study published by the International Communication Association finds that psychological stress was diminished with prescribed media treatments [15]. Additional ways of reducing stress are cited by the National Institute of Mental Health. These include getting regular physical activity, yoga, tai chi, and massage. Keeping a sense of humor, staying connected with family and friends, setting aside time for hobbies and recognizing signs of one's body's response to stress like difficulty sleeping, increased alcohol and other substance use, being easily angered, feeling depressed and having low energy [16].

F. An Engineering Solution to Help Mitigate Stress

There are two types of stress to focus on: phasic and tonic. Tonic stress has to do with background factors uncorrelated to specific times. Pandemic can result in increase in tonic stress as discussed previously due to worry about the economy, spreading of the virus and the impact to one's health. Phasic stress response deals with the onset of stress at a specific time. One's response to a car crash can be considered a phasic stress response, for example. Though both types of stress are important, QB focuses on detection of phasic stress for it's greater ease of detection. The techniques for stress management are also proven to readily deal with phasic stress.

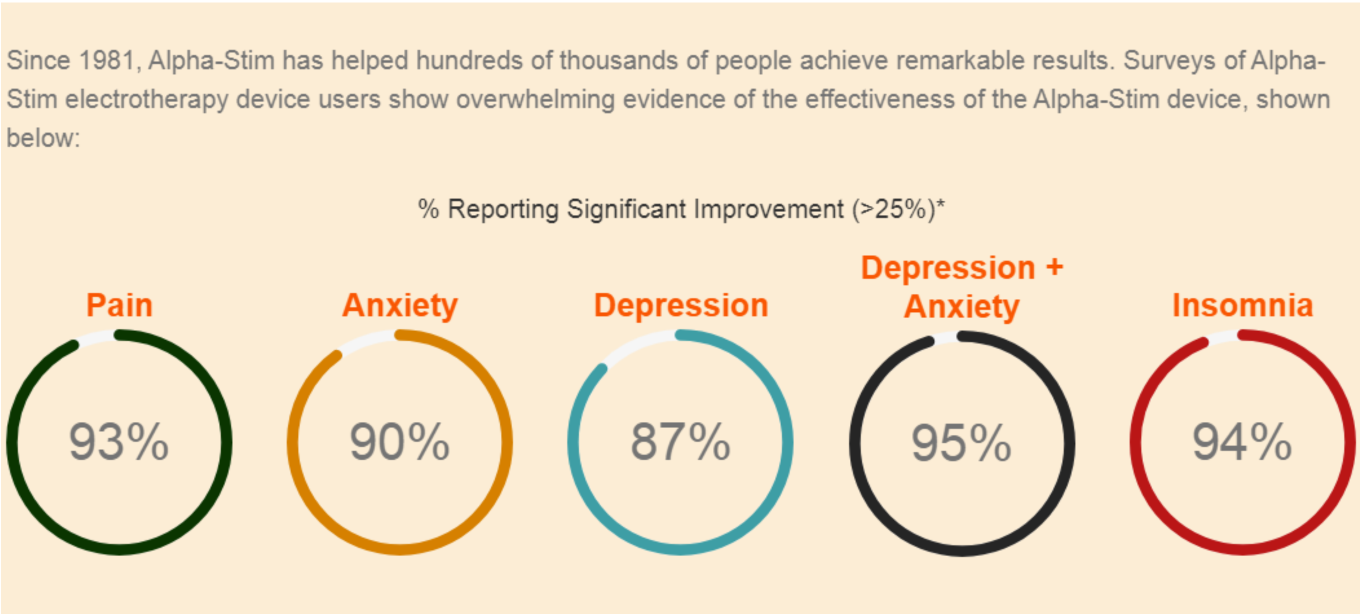


Figure 7. % of patients who reported improvement after using Alpha-Stim [13]

Given the widespread increase in overall stress, it's consequences toward health and wellbeing, and with the knowledge of techniques proven to help, a design to assist in carrying out those techniques while in isolation was developed. Project "QB" (quarantine-buddy) is a stationary social-robotic device that interacts with users over voice or is triggered from a wearable device that tracks stress levels via biometric sensors. These sensors use signals from the body to determine when a phasic stress onset has occurred. When a phasic stress event is determined, QB will perform tasks such as assisting in breathing exercises, making suggestions for exercise, or interacting with outside media to help alleviate stress and related symptoms.

G. Addendum

A Kaiser Family Foundation poll conducted between November and December 2020 found that 51% of respondents across all demographics reported feeling negative impacts on their mental health due to coronavirus. [17] This demonstrates stabilization from the increase in that percentage between March and July of 2020. This suggests that increased stress due to the pandemic and related issues is an ongoing problem.

In a specific case study published by the American Medical Association, "finding suggest that psychological, social, and economic stress related to the COVID-19 pandemic was associated with an increased incidence of stress cardiomyopathy." In the study, an increase of 7.8% in stress cardiomyopathy was found during the months after lockdown measures were implemented as opposed to the same range of dates in the preceding three years. Among others it was also determined that the severity of symptoms and even the average length of the hospital visits were higher among stress cardiomyopathic patients has also increased from before the pandemic. [18] This highlights a very specific avenue of people who have directly experiences the physical consequences of increased stress brought on by the pandemic.

Another survey study found that "the prevalence of depression symptoms in the US was more than 3-fold higher during COVID-19 compared with before the COVID-19 pandemic." [19] This same study not only found that there were less people with no symptoms of depression, but conversely there were more people with more symptoms. [20][21]

The wealth of new information in the months since our last look at the societal problem only enforces the fact that the problem is one that has endured and is affecting more people as time goes on. The symptoms themselves remain the same and nothing our group read led us to determine that a change in our design was necessary. Therefore, we have decided to proceed with the design of QB as planned before with no alterations, additions, or omissions to the previously promised feature set.

III. DESIGN IDEA

A. QB, The Quarantine Buddy

QB (Quarantine Buddy) is a device focused on helping the user manage their stress levels. QB is specifically designed with the COVID pandemic and social distancing in mind. While many are self-isolating and sheltering in place by necessity, access to mental health support may be evaporating. QB is meant not only to measure bodily signals known to vary with stress but to take this information and evaluate the user's current condition. Combined with information gathered via manual input from the user via short intermittent surveys delivered over Amazon Voice Services (AVS), QB will extrapolate various methods for one to manage their own stress levels and make suggestions among these options. QB will also offer various distractions to take the user's mind off the current stressor in the form of short games, calming audio or visual media content (music, cute baby animal videos etc.) and even assist with soothing breathing exercises.

Originally, QB was to sample the user's heart rate (HR), heart rate variability (HRV) and galvanic skin response (GSR), to not only establish the average state of the user but to detect variations in the norm to detect moments of stress (MOS). As the project progressed, an algorithm was discovered requiring only ST and GSR. In the final prototype, only ST and GSR were used to calculate the stress score. These measurements are taken via a device worn on the hand. Information is transmitted wirelessly via Bluetooth to a main robotic module that polls for signals indicating stress.

The main body of QB is where the rest of the functionality takes place. QB will store the measurements taken from the user and perform operations to determine its reaction to the user's various states. These reactions are designed using Amazon Web Services and Alexa Voice Services (AWS/AVS). QB will have an appearance that is roughly humanoid to perpetuate the notion of a companion. He will be stationary but will have moving limbs symbolic of arms and will be able to move his head slightly. These movements add personality to QB. Figure 8 illustrates QBs overall design functionality in terms of its individual features and how they interact with the other modules.

B. Addressing the Problem

One of the most difficult things of sheltering-in-place and social distancing during a pandemic outbreak is the lack of companionship and emotional support. QB looks to fill in that gap by assuming the role of a comfort companion to those suffering from severe stress while confined to their solitary living spaces for a large percentage of their time. QB will assist in self-regulatory stress techniques proven to help relieve and manage stress in addition to providing small distractions also known to contribute to a healthy mind and less stressful state of being.

C. Under the Hood

The wearable stress monitor will incorporate ambient light sensing technology to monitor heart rate. A galvanic skin response sensor and a temperature sensor are also part of the sensor array. Bluetooth will also be utilized for communication between the wearable measuring device and main robotic module. Electronic current sensing technology will be used to implement the EDA feature.

The robotic will use servo motors to implement preprogrammed gestures based on user and own responses. Simple LEDs can be used as eyes and both can be programmed by microcontrollers such as an Arduino or Raspberry Pi. In addition, Amazon Web Services (AWS) will be leveraged to provide online media interaction and voice I/O. A touch screen will be included for visual questions and activities. All these features will be implemented on one or more microcontrollers.

D. What Makes QB Unique?

While interactive automated companions are nothing new, with applications ranging from children's toys to more high-tech models that can perform more complex tasks, QB exists solely to alleviate stress and that is what makes it unique. While it will have limited functionality in terms of entertainment, more accurately perhaps for distractions, it will be monitoring bodily signals

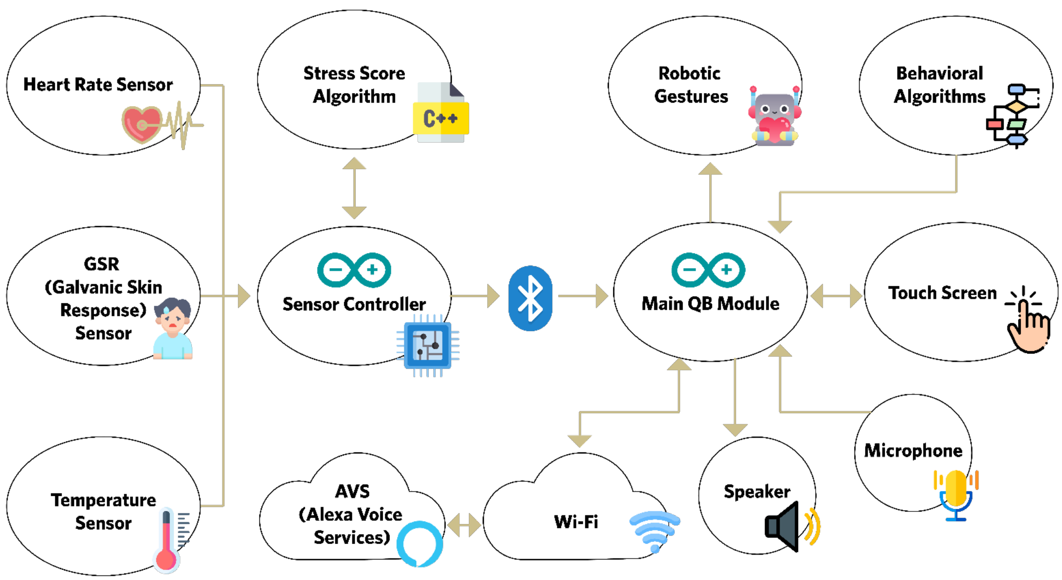


Figure 8. Flow Diagram of Proposed Design [30]

and taking in other information on user's stress levels through other means like asking questions and using these to determine how it can best help the user to mediate stress levels. The function of QB solely as a stress companion tailored specifically to those confined to their homes for a disproportionate amount of their time is what makes QB such a novel idea and a unique approach to stress management.

Several technological solutions exist for stress detection and treatment that are similar to QB. The *Alpha-Stim* device, the Fitbit *Sense* and *PARO* are three examples of the different approaches to mitigating stress. Alpha Stim is an electrotherapy device. Cranial electrotherapy stimulation (CES) delivers a "natural level of microcurrent via small clips worn on the [users] earlobes... to stimulate and modulate specific groups of nerve cells [13]." This type of approach is non-invasive but is different from the approach proposed with QB in that it requires the user to monitor their own level of stress and apply therapy when needed or desired. Fitbit products are marketed as a wearable health and fitness tracker that track things like HR and HRV. High end models like the Fitbit Sense are also capable of monitoring stress by adding EDA measurements and employing algorithms that determine a daily stress score [21]. This product exemplifies the approach used by other manufacturers such as Apple and Garmin with their respective products, the Apple Watch 4 and the Garmin Vivosmart 4. Both are wearable products that serve as a standalone tool to monitor stress while providing. Interactivity is available in the form of applications that mimic breathing exercises on all three products, similar to what QB will provide.

Another approach to the management of stress while in isolation is PARO, an advanced interactive robot developed by AIST. PARO uses various sensors to behave like a trainable pet. It can learn a new name and respond to a use's physical interaction, like petting or training. Currently, PARO is being used in more than 30 countries around the world in helping patients suffering from diseases like Alzheimer's, Autism, PTSD (post-traumatic stress disorder) and other forms of dementia [22]. A former astronaut on the ISS said, "based on my experience, I think that PARO would be very popular with the crew [on long space expeditions]. PARO would improve emotional support, help with isolation issues, and improve crew bonding."

One of the main differences between the approaches listed and that of QB is the combination of the listed approaches. QB actively monitors a user's stress levels, unlike the Alpha Stim. When a moment of stress (MOS) is detected, QB actively tries to assist the user in alleviation of that stress. Unlike the wearables mentioned previously, QB can physically interact, like PARO, as a companion while keeping the user in isolation.

E. Necessary Resources

For the wearable monitor, a sensor for both heart rate and galvanic skin response are needed. The heart rate sensor will be based around ambient light sensing technology. The GSR sensor will use an electric current sensor. The ST sensor uses infrared light to generate a voltage signal. All sensors will output analog signals which are then amplified and filtered to appropriate levels differentiable to our chosen microcontrollers analog to digital converters (ADC). The microcontroller will communicate the data via a Bluetooth module to the main QB robotic module. The Arduino Uno will be used for the initial proof of concept. This will require the use of the Arduino 1.8.13 IDE for programming the various algorithms to gather and send data from the sensors to QB. The language used for this IDE is C++.

The robotic will need a separate microcontroller device (either Arduino or ESP-32). It will use Bluetooth for communication with the wearable and interaction with online media. Other I/O will be with an Echo Dot speaker/microphone and home laptop for web interface. Amazon Web Services (AWS) and Alexa Voice Services (AVS) will be used to provide a better companionship experience. AWS and AVS will help the design by providing valuable features such as access to media files, voice-interactions, and real time responses. Preprogrammed gestures for arms and head will make use of several servo motors to provide a more comforting companion.

Project QB will be undertaken by Senior level undergraduate students who are themselves self-isolating. Therefore, lab space is required to be in home office environments. University lab equipment was not available; therefore, electronics benchmark equipment was limited. The functions of the digital Analog Discovery 2 will serve as the primary electronics laboratory measurement equipment. This device offers DMM, + and – 5V power supplies, oscilloscope and function generators and interfaced with Digilent® WaveForms software version 3.12.2 for Windows 10 (10.0).

Primary consultants for the project are California State University, Sacramento Professors Neal Levine and Russ Tatro.

There was little to no travel required to carry out this project and the designers themselves served as their own technicians. Soldering and sewing of the final wearable sensor was carried out with a Hakko® soldering iron on perforated PCB board. 3D printing done for the limbs of QB and assembly only required super glue.

All parts were ordered through online vendors Digi-Key or purchased from local electronics retail stores.

F. Features of this Design & Metrics Used to Determine Sufficient Implementation

- **Biometric Monitoring**

Readings of HR, GSR, & Temp provide sensing ability that is able to detect a phasic stress response of the body during a 15-minute test in which a user undergoes stressful a stressful situation: difficult examination, heavy traffic etc.

GSR is implemented using the Grove GSR sensor. Resistance between the fingers is sampled and used as input to an instrumentation amplifier built using LM324 op-amps operating at 3.8V. The value of GSR sampled every second using a 10-bit ADC. This value is stored for 20 seconds and is used by the stress detection algorithm to help calculate the magnitude of a phasic stress response.

ST is monitored with an infrared ZTP-115 senso r. The sensor operation is based on the Seebeck effect, in which a voltage difference is established between the junction of two different metals when heat is applied. Infrared light from a user's skin is the source of heat in this case. A basic feedback amplifier was used to boost this voltage difference by about 1000V/V. This signal was input to a 10-bit ADC and sampled every second. The value was stored for 20 seconds and used along with GSR to calculate a stress score every second.

Implementation of this feature requires the ability to read a 0.1 degree C difference in temperature with the Arduino's 10-bit ADC.

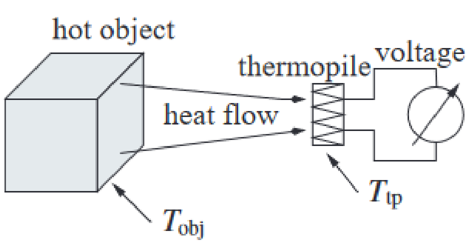


Figure 9. Diagram of Seebeck Effect [31]

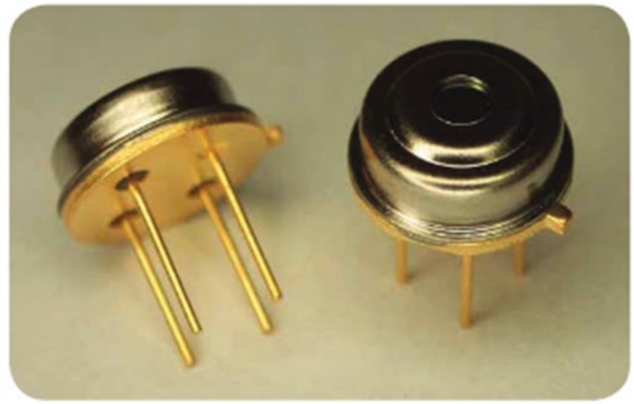


Figure 10. ZTP-115 Infrared Temperature Sensor [31]

- **Bluetooth Wireless Communication**

Several pre-designed solutions exist to implement this feature. The ESP-32 board has built in Bluetooth antennas and a library of software that allows development in the Arduino IDE. The HC-05 is also designed to work with the Arduino family of microcontrollers. The latter solution was eventually used implement Bluetooth connectivity between the Arduino Lilypad connected to the wearable stress monitor and the ESP-32 board that connected to the stationary robot.

This feature was considered fully implemented when 99% of test signals from the wearable are detected by the robotic module.

- **Robotic Movement**

Performs defined movements when signaled by the user. Initiation requires voice activation by the user.

- **Voice User Interface**

Feedback between user and QB via voice services is established and provides appropriate content to user stress state.

This feature has endless development potential, but for this project is considered sufficient when a webpage displays and updates content based on a user's voice responses to questions regarding their state of wellbeing.

G. Biometric Monitoring

The entire design relies heavily on the sensor array that will be worn by the user. The array consists of three sensors: one for heart rate, one for galvanic skin response, and one for temperature. These sensors will be interfaced to an Arduino board (Lilypad), that will calculate a stress related score based on an algorithm that uses GSR and ST. Software for this feature is minimal. The algorithm for calculating the users stress response is programmed in C++ code written in the Arduino IDE (current version 1.8.13). Arduino supports C and C++. The decision to code in C++ is to take advantage of the language's object-oriented characteristics as opposed to the purely procedural nature of C. While this is hardly relevant to the code controlling the sensor array, it will be extremely helpful in processing the data gathered from the sensor readings in the main module.

A heart rate sensor that senses ambient light will be used to detect a user's pulse. The signal of the sensor will be used to calculate HR and HRV. A GSR sensor will be implemented. This sensor detects current on a user's skin. When the skin is moist (as when sweating) signal amplitude will increase. Amplification and filtration is needed and was implemented using discreet circuit components on a breadboard for the initial prototype and proof of concept. This was eventually transferred to a perforated printed circuit board. The sensor will be used to monitor a user at a "resting" state to establish baseline GSR. Known stressors will be applied to a user and GSR will be recorded.

Zach and Jayson will be overseeing the development of the sensor array. Zach will focus more on the physical circuit design. This includes noise filters, and high/low pass filters as needed and any necessary amplifiers. Jayson will be more in charge of the writing the C++ code that will control the sensor array and handle the taken measurement. Design, construction, and coding of the sensor array will take an estimated 25 hours to fully implement. Jayson brings programming experience along with knowledge of computer interfacing, data structures, and algorithm analysis in order to bring about successful function of this feature. Zachary will apply knowledge of signal amplification and noise filtering techniques that will come in handy when dealing with biometric signals and in the physical design of the sensor array itself.

To ensure that this feature is implemented completely and successfully we must verify that the sensors not only take their respective measurements at the appropriate defined intervals (as determined by the controlling C++ code) but also that these readings are accurate. To test for accuracy, we will compare the readings of our sensor array with those taken from commercially available devices already accepted to be sufficiently accurate with the goal of 90% accuracy.

H. Wireless Communication

Wireless communication between the sensing device and the robotic module will be carried out via Bluetooth. Both the wearable and the robot will have a Bluetooth transceiver. For the initial prototype, the HC-05 module will be used. Arduino IDE and C++ will be used to develop the software protocol that handles communication between devices. For the prototype, 20 hours have been set aside to implement this feature. This feature will be considered fully implemented when 99% of test signals from the wearable are detected by the robotic module under when operated within the distance and power ranges specified in the modules data sheet. This feature is among the most foreign to the skillset of the group as a whole and therefore will most likely require a little help from everyone to implement. Skill in the application principles in RF design and communication to will be beneficial.

I. Robotic Movement

Robotics is an abstract implementation of sensors and actuators and using their calculations to interact with the environment. The QB bot will make use of servo motors to make the head and arms perform gestures. These gestures will be preprogrammed and will rely on the phrases used by AWS. Servo motors allow for connected components to rotate. The connected components will be arms and a head which can have simple shape and be made of wood or 3d printed. A simple use of LEDs can serve as eyes. These components will have to be implemented on a microcontroller and connected by a breadboard in order to be powered by around 9V for the servos. The microcontroller can be an Arduino or Raspberry Pi. More interaction will include a touch screen that will have inquire the user for questions and recommendations. This robotic module will mainly react to the readings from the stress sensors and speech using AVS. This will be a stationary robot since we are concerned with support inside the home during social isolation. Victor will be spearheading the movement features of the robot, with his experience and interest in the field of robotics being particularly suited to this task.

J. Voice User Interface

Amazon Web Services (AWS) is a secure cloud services platform which provides users with multiple assets and utilities such as storage, cloud computing and media delivery. For this project, the team decided to lever the power of AWS to execute multiple features that will be implemented into the design. One of the biggest features that will be leveraged via AWS will be showcasing video, music, or photos via a display. A storage service of AWS that will be used is Simple Storage Service (S3). S3 stores video, music and photos as objects. Objects are placed into repositories otherwise known as "buckets". S3 allows for easy control over the directories and makes remote access to objects and buckets be secure. In order to execute specific requests, AWS' Lambda will be used. It is a cloud computing engine that will be used to generate code that can control specific servers from AWS. For example, Lambda can be used to respond to changes in specific S3's buckets and pull/push data to an Amazon Database. "Lambda performs all the operational and administrative activities on your behalf, including capacity provisioning, monitoring fleet health, applying security patches to the underlying compute resources, deploying your code, running a web service front end, and monitoring and logging your code." [23]. There are many available databases within AWS console. One of them being DynamoDB which is a simple scalable database that allows user to implement their own data tables and automatically update them without any risk of data corruption. This type of database can be used for the collection, verification and validation of vital measurements that will be conducted during the entirety of the project.

One of the major features of this design will be to implement a voice interaction and feedback. For this, the team decided to implement Alexa Voice Services (AVS) which contains a voice user interface and a natural language processing engine. The voice user interface will help bridge the gap between the bot and human interaction. While the natural processing engine will decode specific key phrase and words to understand the task at hand. Alex will be responsible to configure the infrastructure behind the use of AWS and AVS for this design. *Figure 11* shows the flowchart of how commands will be processed using AVS and AWS.

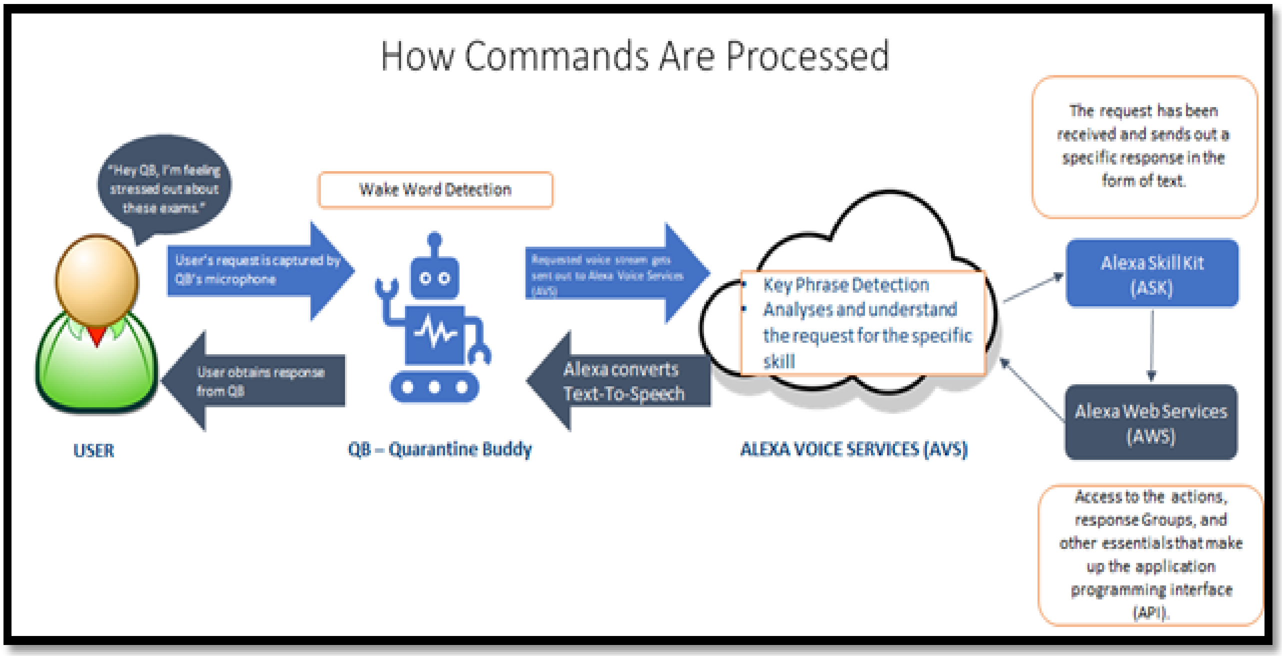


Figure 11. Flow Diagram of Amazon Web Services [30]

IV. FUNDING

All expenditures were directly out-of-pocket from all the members of the team. We did receive help with the fabrication of certain components used for the final prototype that certainly saved us a great deal of money. However, all costs were fronted by the team. Here we will provide an itemized list all parts used in the final QB prototype, many more parts were purchased for design consideration and testing throughout the entire research and development phases. These include, but are not limited to different types of microcontrollers, extra microcontrollers just in case, replacement microcontrollers when a desired feature was absent on a board we were currently using, extra batteries, and different kinds of temperature sensors.

ITEM	AMOUNT	PRICE/UNIT	TOTAL
LilyPad Arduino Simple Board	1	$24.95	$24.95
LilyPad FTDI Basic Breakout – 5V	1	$18.95	$18.95
Lithium-Ion Battery – 110mAh	1	$4.95	$4.95
Grove LM324 GSR Sensor	1	$9.90	$9.90
ZTP-115 Temperature Sensor	1	$6.60	$6.60
Pulse Sensor	1	$24.99	$24.99
LM324 Op-Amp	1	$1.29	$1.29
ESPwroom 32	1	$10.00	$10.00
Amazon Echo Dot	1	$19.99	$19.99
HC-05 Bluetooth Module	2	$10.57	$21.14
DC Servomotor	7	$5.95	$41.65
Hand strap as housing for sensors	1	Hand crafted, Negligible Cost	$0

Housing for QB body, limbs, & head	9	3D printed by personal acquaintance at no cost	~$50
TOTAL			$214.42

Table 1. Bill of Materials

V. PROJECT MILESTONES

A complete timeline detailing the start date, time required, and completion date is shown in the Gantt chart and major task dependencies are found in Appendix G. This chart also shows the major project milestones that were used to gauge the progress of QB's development. These are highlighted in yellow in the PERT chart.

A. Team Societal Problem

The first major milestone of project completion was the team's definition of the problem to address. Each of the four members of the Loneliness Avenger's team researched and developed their own view of a societal problem with an engineering solution. While all problems presented were unique, the majority converged on addressing the effects of isolation during pandemic, and so a single team idea was refined into what has led to development of the Quarantine Buddy.

B. Design Idea contract

The second major milestone was developing the feature set for QB in the design idea contract. Several group discussions and brainstorming sessions led to the development of the feature set QB. This contract was the basis for which design completion would be determined.

C. Work Breakdown Structure and Timeline

The feature set of QB is diverse and requires a great deal of planning to manage progress. The breakdown of each major feature into a set of tasks with a primary overseer and the implementation of a timeline to schedule development over the course of two semesters was the next major milestone of this project.

D. Project Integration

Although the work breakdown and timeline are helpful in identifying potential barriers to completion, roadblocks to fulfilling the design idea contract are likely to become apparent when integrating the features of QB. A risk assessment will be performed to help anticipate these roadblocks and a provide the team with the flexibility to overcome them in a timely manner. The full integration of all prototype features represents the largest hardware development milestone.

E. Technical Review & Presentation

Final testing and debugging will lead to the completed first prototype of the Quarantine Buddy, enabling its presentation. This presentation will be the final milestone of the first semester in the timeline.

VI. WORK BREAKDOWN STRUCTURE

A. Biometric Monitoring

The wearable sensor array takes the user's biometric readings and is, in effect, QB's eyes and ears. The sensors actively monitor certain vital signs of the user in order to detect when a candidate MOS occurs. QB will then process these readings according to our MOS detection algorithm to determine when such an event takes place. The breakdown of the sensor array and the processing of the information gathered is as follows.

1. MOS Detection Algorithm

First, we had to research instances of studies done using similar methods of detecting a phasic MOS using biometric readings. Our main source of inspiration was "Detecting Moments of Stress from Measurements of Wearable Physiological Sensors" published by the National Center for Biotechnology Information. The reliance on only two signals for (GSR and ST) and stated 80% accuracy of detection led to its selection for implementation.

After learning proven methods of MOS detection, the next step is to create a flowchart of the logic the algorithm. This allows visualization of the logic at work and to follow the algorithm step-by-step.

The flowchart of the MOS detection algorithm is shown in Appendix C. The algorithm shows that 5 rule weights are to be calculated based on the rise time of GSR, the time difference between the GSR minimum and maximum values, slope of the GSR, temperature decrease soon after a determined MOS event and number of MOS events within 10 seconds of each other. These rule weights follow the description of the NCBI study cited earlier. *Figure 12* allows visualization of components of the stress score based on a GSR peak. The final stress score is calculated every second according to the description.

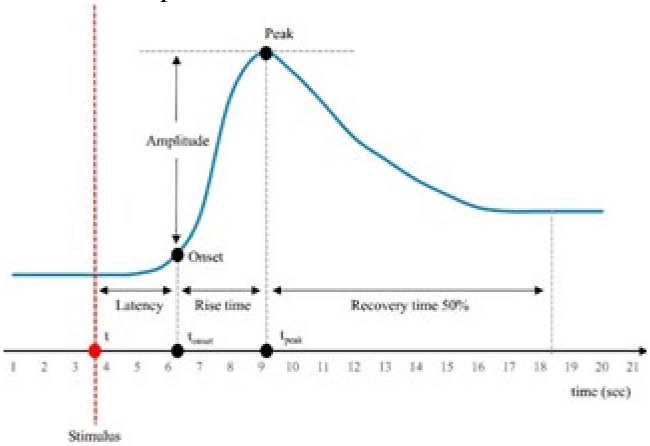


Figure 12. Example of GSR spike and components of the curve used in determining magnitude of stress response [32]

Each rule is calculated as follows:

R1. [gt:gt+5]′ > 0

Where gt is the value of GSR and gt+5 is the value of GSR 5 seconds afterward. A full score of 2 is allocated if a net increase is detected in this time interval. If a net increase is detected for a period longer than 5 seconds, half the rule score (1) is used. Otherwise, the rule is scored 0. If the rule is satisfied, a candidate MOS is recorded and the time value (t) of this MOS is used in the remaining rule calculations.

R2. [Tt+3:Tt+m]′ < 0

If temperature (T) decreases for 3 or more seconds beginning 3 seconds after t of the candidate MOS, the full rule score is allocated. If the decrease occurs between 2 and 6 seconds after t, half the rules score is allocated.

R3. tpeak − tonset ≤ 5 s

If GSR rise time is less than 5 seconds, full score is allocated. If risetime is greater than 5 seconds, half the score is allocated.

R4. (gpeak − gonset)/(tgpeak − tgonset) ≥ 10°

The slope of GSR was used for calculation of Rule 4. Documentation did not reveal the values of GSR used to obtain the threshold angle used for maximum rule score. The equation was used as is with the GSR values obtained, however tuning of the associated maximum rule weight or adjustment to the GSR amplification should be performed to account for the different scale of GSR used in these results versus those obtained in the development of this algorithm. . In it's present state, a maximum score of 2 is assigned when the condition is met.

R5. tMOSi+1 − tMOSi > 10 s

A maximum rule 5 score is used if more than one candidate MOS is detected within 10 seconds. Otherwise the score is 0.

Each maximum rule was given a weighted value. These were R1, R2, R3, R4, R5 = {30, 25, 20, 15, 10}. These values are somewhat arbitrary and prioritize the lower numbered rules. Tuning of these values, as discussed previously, could help increase the accuracy of stress detection.

The final stress score, SC, is the sum of the product of each rule score and the associated rule weight, Wn:

The idea for programming the algorithm into an FPGA was one possibility, since it lends itself to the parallelism of the calculation, but since the speed of output required is slow (1 Hz) it was decided to continue development in the Arduino Uno microprocessor. A proper delay time is added after each loop through the algorithm to ensure 1 Hz operation. A listing showing the implementation of the algorithm in C++ with the Arduino IDE is shown in Appendix C.

After translating the flowchart logic into working C++ code, it was tested and debugged using the sensor array. Once it was established that the sensors take proper measurements from the user at the determined sampling rate and those readings are correctly processed to detect when a MOS occurs the sensor array module was complete.

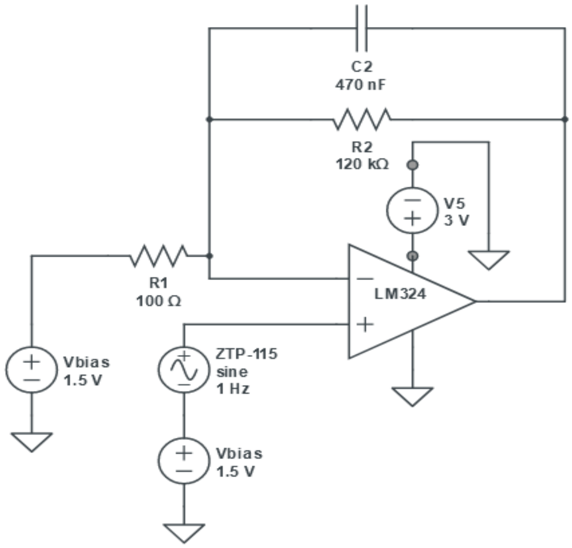


Figure 13. Temperature Sensor Amplification Circuitry [30]

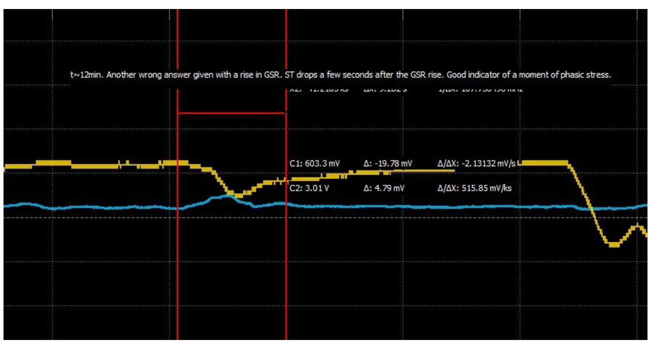


Figure 14. GSR measurements (Yellow) vs Skin Temp. (Blue). The area between the red lines indicates the introduction of a stressor, resulting in the following spikes and dips in the respective waveforms [30]

Problems arose when testing of the algorithm began, revealing problems in the WBS. For one, the original temperature sensor was discovered to be too slow to respond to changing temperature as well as too insensitive. The original temperature sensor was the DS18B20. This sensor integrated seamlessly with the Arduino environment, but testing in a live environment revealed output to be too slow. The resolution was also not deemed high enough to resolve small changes (~.1 degree C) in user skin temperature. Since this sensor output a serial value of the data, the signal could not be amplified to increase the resolution. Switching to an infrared sensor allowed testing and debugging of the algorithm to finish.

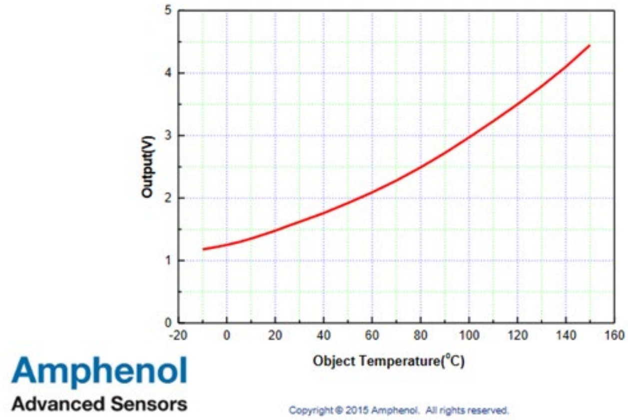


Figure 15. Voltage as a function of Temperature for the Infrared Sensor [31]

2. Building the Sensor Array

Since the sensor array provides the input needed for QB to determine when the user is in need of emotional support, the accuracy of the readings is vital to the effective functionality of QB. The first task was to ensure the accuracy of each individual sensor of the array and to understand its inherent traits in order to effectively program its behavior within the entire array. Firstly, the Pulse Sensor was explored to detect the user's heart rate. The complete picture of the user's heart rate comprises of two separate values, beats per minute (BPM) the heart rate variance (HRV). The manufacturer of the Pulse Sensor provides an open source library of functions compatible with Arduino. These functions were used to experiment with the Pulse Sensor until satisfactory performance was achieved and the stable behavior was observed. This step involved writing C++ code for Arduino, testing the accuracy Pulse Sensor against a medical heart rate monitor to ensure accuracy, and debugging the code as needed.

Secondly, the same process is gone through with the temperature sensor. Initially we used the common LM35 which outputs and analog value of 10 mV / C°. First the characteristics of the sensor were determined through testing and experimentation before the final C++ code was written. The C++ code controls the rate at which the temperature is sampled and stored and in the case of this temperature sensor also handles the analog-to-digital conversion and converting Celsius to Fahrenheit. It was discovered early on in this process that an unacceptable amount of precision was lost when converting Celsius to Fahrenheit on top of converting the analog signal to digital. Thus, it was decided to abandon the LM35 and swap in the digital DS18B20. Once we went through the same testing and experimentation to learn the behavior of the new digital temperature sensor, we found our readings to be much more stable and precise without having to convert analog voltage to a 10-bit digital value. Due to the nature of the new temperature sensor, which requires much less power the entire circuit ended up being much more stable as well.

The algorithm testing and debugging continued with the DS18B20. This sensor was deemed too slow to respond to the rapid and subtle changes in skin temperature. The MOS detection algorithm relies on changes in skin temperature, rather than the absolute value of skin temperature. The DS18B20 functions as an ambient temperature sensor. For a user to trigger temperature changes, the sensor needs to be housed close to or encapsulated by the body i.e. in the user's fist. The sensor outputs digital signals that the Arduino's predesigned library interprets as temperature readings. This means that the sensitivity cannot be readily increased.

These facts warranted the change to analog circuitry built around the infrared ZTP-115 temperature sensor. The specifications for this sensor are listed in the Introduction. Amplification circuitry was required to boost the signal to values differentiable by the 10-bit ADC of the Arduino UNO. The sensor is equipped with a thermophile, which is used to detect infrared and output a linearly proportional voltage. It also is equipped with a thermistor for temperature compensation. The thermistor is not used since the algorithm relies only on the change in temperature. Future designs should include this compensation to help stabilize the ST readings, however. The circuit that was finally employed is shown in Figure 13. Figure 15 shows the sensor voltage/temperature output curve.

Lastly, the Galvanic Skin Response sensor was experimented with and its behavior learned. This device was a predesigned Grove GSR sensor module. It uses metal leads connected to a user's skin: one on the index finger and one on the middle finger of the same hand. One lead is connected to the ground of the system. The potential difference between the two leads is read into an instrumentation op-amp that is built around the LM324. The potential difference is proportional to the resistance between the leads: an increase in the voltage reading means an increase in the resistance between the leads. GSR, therefore, is the inverse of the output voltage value. When the voltage of the GSR sensor output rises, GSR is interpreted as falling and vice versa. Once it was determined how to interpret the sensors output for our purposes its controlling C++ code was written and debugged until satisfactory performance was achieved.

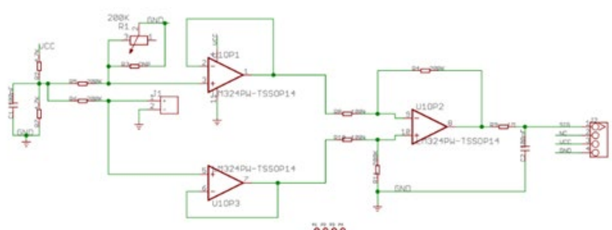


Figure 16. GSR amplification circuit [32]

Combining all sensors into one circuit required more debugging of the entire code, as some of the sensors needed more turnaround time after its measurements were sampled so the entire circuits sampling rate was adjusted. Once the complete sensor array was working properly, that is accurately and stably, the sensor array is complete.

B. Wireless Bluetooth Communication

Implementing a wearable device warranted the need for wireless connectivity. This was accomplished again using predesigned modules: the HC-05 Bluetooth module. This device is easily programmable and works easily with the Arduino

Family of Microcontrollers.

The HC-05 programming requires AT commands to be entered using the Arduino Serial Monitor. First an empty sketch must be loaded into the Arduino MCU. To program the device, the pins are connected TX->TX and RX->RX. AT commands can then be entered to establish the Master/Slave relationship, Baud rate and passwords if desired.

The Baud rate of 115200 was chosen in order to interface easily with other IoT applications discussed later.

HC-05 modules communicate using the TX and RX pins of the Arduino. Data is output the Serial Port Protocol (SPP). For this design, 1 start bit is used (Active Low) followed by 8 data bits and 1 stop bit. Testing was done with the AD2 oscilloscope to verify this.

After pairing, the HC-05 modules are connected with TX->RX and RX->TX when connecting an HC-05 to an Arduino. If connecting an HC-05 to an ESP-32, the connection remains as before. The onboard Bluetooth of the ESP-32 development kit was unused.

1. Establish Communication Protocol Between Two Arduino Boards

To enter AT commands into the HC-05 module, the unit must be connected to power. An empty sketch is then loaded into the Arduino microcontroller. After this, the RX and TX pins of the Arduino are connected to the RX and TX pins of the HC-05. The AT commands used to interface the two microcontrollers were entered into the Serial Monitor of the Arduino IDE. These commands establish the "Master/Slave" relationship, set the BAUD rate and the serial communication protocol.

2. Set Up Bluetooth Communication Between QB and Sensor Array

Testing was carried out to check the reliability of this communication link. A program outputting integer values representing the stress score at 1 Hz was uploaded into the MCU of the wearable module. A program polling the receiving pin (RX) was uploaded into the stationary MCU. Output was tested to verify the output was received at the input at the correct frequency and in the correct order. No parity bits or CRCs were used.

C. Robotic Movement and Gestures

In order to give QB a personality to easier facilitate its role as a stress relief companion it was decided to give him robotic functionality to supplement his vocal interactions. This will serve to give QB more of a personality. Victor has spearheaded the programming of the actuators and servos for the robotic movement.

1. Programming the Gestures

The programming of the robotic movement is fairly straightforward so far. We have a small amount of gestures for QB to perform. These gestures are a wave of the hand as a greeting, nodding of the head, and a fist bump. The coding for these is done and debugging and tweaking will most likely need to be done once we start to build QB and the arms and head are all put together in its final form. Once that is done, we will be able to see the servos functioning in their actual positions for debugging to be done successfully.

2. Casing for the Limbs and Head

Once the internal hardware for QB is finalized we will be able to establish the form it will take. This includes size and the location of the hardware relative to each other. Once this is established, we will need to use a 3D modeling program to design the outer casing of the QB unit. We have a source of 3D printing lined up for when we are ready for production.

3. Synching Gestures with Alexa

This is what we foresee as being one of the most challenging parts of the project. Alexa Voice Services will be providing the vocal interactions with the user so if any gestures are to be made at the appropriate time it will be necessary to synchronize them with what QB is saying. Once we establish full Alexa functionality, we will be more prepared to approach this portion of QB's.

D. Voice User Interface

Going into this project we foresaw this portion of the project as being must less involved than it has proven to be. Alex has previously programmed Alexa functions before at his internship with Intel. Since his last experience doing this, Amazon has seemingly ported their Alexa Voice Services to a different platform making programming it very different. Once we realized Alex was being unfairly overworked we decided to all learn the basics of Alexa Voice Services programming in order to have more hands on deck given that other QB features are at a satisfactory point and are more or less waiting for this feature to be implemented before we can proceed anyhow.

1. Create Alexa Skill

An Alexa Skill is basically an application within Alexa. The entire function of QB will be encapsulated entirely within one Skill. Part of this process is to create the *invocation name*, which are the words that need to be spoken by the user to activate the desired Alexa functionality. After the invocation name is created the *intents* must be written. The intents are the actions Alexa will take once the Skill is activated by the invocation name. Dialog Confirmation when the intents are entered must also be written. We must also write *slots* which are basically variables within Alexa's utterances. Slots cannot be used within built-in intents which is part of the reason why custom intents must be written. To properly implement slots, we must write the variables, in this case words, that will fill these slots at the appropriate time. Slot confirmation must also be accounted for. Slot validation rules must also be written.

2. Create Lambda Function

Lambda is the computing service provided by AWS. This is what runs the code in response to the programmed events and automatically manages the underlying resources during Alexa's functioning.

First the Lambda function must be formatting into node.js language. Then the handlers, functions, and sessions that serve as vocal output to the user must be written. We then need to create and validate an additional Lambda function responsible for reading and writing to our planned database that will hold various forms of audio/visual media to help calm the user. To accomplish this, we will link the Alexa Skill Kit as a trigger to the main Lambda function. In other words, a Lambda function called by another Lambda function. We will then create another Lambda function that controls the reading and writing to and from the database.

3. Create the Database

Amazon provides a proprietary NoSQL database service called DynamoDB. To use DynamoDB for our database we will create a DynamoDB table. Then we will set up permissions in order for our Lambda function to access it for read and write operations. Then the tables attributes and data value types will need to be set up.

In order for us to use the database in QB we need to set up the AVS API to handle output to a web client that will be used for the touch screen interface with QB. First, we will need to create an API using API Gateway Console within AWS. Then we'll need to link the API to the Lambda function that will read values from the database. Part of deploying the API will be to enable cross-origin resource sharing.

E. Visual Display

QB's visual display will be displaying a locally hosted web page instead of creating our own proprietary GUI. This was deemed to be easier to integrate with Alexa than the other option.

1. Creating the Website

The website facilitating user touch interaction will be created using codepen.io and will be written in JS, CSS, and HTML. These are the standard languages used for website design and will allow us to create a functional webpage, style it how we see fit for our aesthetic vision and enable the interactivity for users to select options and answer queries.

The website will access the DynamoDB database to display the various forms of media held within. This will include relaxing music, breathing exercises, and even some physical exercises. This will involve creating JS functions to access our proprietary API to display the correct form of media accordingly.

VII. RISK ASSESSMENT

Development of QB will require pathfinding and completion of several critical features. Various technical and systematic risks are at play to hinder completion of these features. The technical risks can be characterized as specific and broad in nature. The systematic risks are associated with the team's busy school schedules, quality of life and the unique development environment brought on by social distancing.

A. Specific Technical Risks

The wearable module will make use of a battery. This has the risk of overheating, leaking, or exploding. Mitigation, should this risk become manifest, will involve engineering power regulation solutions. This risk is deemed low, but its impact is high as it is desired to have the wearable operate wirelessly at low risk to the user.

The main robotic module has a low risk associated with powering the seven servo motors that control movement. Impact is deemed high if this risk is manifest. Mitigation will involve engineering power supply and electro-mechanical solutions.

Proper implementation of the Bluetooth protocol is deemed a medium specific technical risk. Impact is low, as there are several other means of communication between the main and wearable modules. One of the technical risks will be implementing Bluetooth connectivity between the final prototype wearable (if it is implemented on the ESP32 development board) with the Arduino Uno that controls the main module. Mitigation of this risk can be accomplished by using an Arduino type board that has Bluetooth integrated.

Several specific technical risks are associated with completion of Alexa functionality. Of medium risk levels are construction of a database to hold stress scores (low-medium impact), interaction of the lambda function with the database (high-impact), and building an Alexa skill with a conversational tone and interface of AVS with the Arduino/Raspberry Pi (low impact) Mitigation will involve pulling of all available resources to complete these tasks.

B. Broad Technical Risks

Accurate interpretation of biometric signals is a technical risk that covers a broad scope. The risk factor is deemed medium while the impact on the project is deemed high. Without proper detection of MOS, QB cannot carry out functionality to alleviate stress and related symptoms at the critical moment needed by the user. Mitigation of this will require reengineering of the MOS detection technology. This may include use of more accurate sensors, employing different biometric sensor types, research and development of different detection algorithms, higher capability multi-core processors and redesign of testing methodology. The order of these mitigation techniques will be highly situation dependent.

Execution of Alexa functionality is deemed a medium level specific technical risk. Impact on the project is deemed critical. Mitigation of this aspect of risk will require pulling all team hands into the development of this feature. Multiple members of the team will need to learn the programming environment and assist the main developer in execution. Worst case scenario will be to deploy the prototype while Alexa capabilities remain in development. QB can perform several proof-of-concept functions without, but the impact without AVS and AWS will be greatly diminished.

C. Systematic Risks

Busy school schedules were deemed a low risk to project completion. All members of the team have demonstrated their ability to juggle multiple assignments to a high level of execution. Impact to project completion is deemed high, however. Mitigation involves scaling back features of QB to arrive at a deployable prototype by the deadline.

The risk of developing during social distancing was deemed as high. The impact of this risk was medium. Social distancing, as discussed, has led to the closure of many of the facilities that were originally needed for development. University lab space was restricted, meaning use of oscilloscopes, power supplies and general electronics equipment was limited. Mitigation of this risk involves setting up lab spaces at the homes of individual team members. Purchase of a small work bench, inexpensive digital oscilloscopes, a soldering iron, and station equipment were all done to mitigate the impact this risk had on development. Use of Discord streaming software can be used to establish meetings and work sessions between team members that follow social distancing guidelines.

The availability of parts required for building the prototype is deemed a high-level risk because of the likelihood that pandemic would affect delay in shipping. Impact on development was deemed low. Several parts exist in the inventory of the team members from previous projects, enough to build a proof of concept. In addition, team members were to proactively order parts as early as possible to mitigate the effect long delays in manufacturing and shipping would have.

RISK ASSESSMENT MATRIX

PROBABILITY

5					
4				Ineffective MOS detection algorithm	Unable to implement Alexa
3	Unable to implement Bluetooth				
2				Problems with robotic movement	
1				Issues with battery in wearable	
0	**1**	**2**	**3**	**4**	**5**

IMPACT

VIII. Design Philosophy

QB was designed to help mitigate the increases in stress brought on by the COVID-19 pandemic. This stress was found to be a result of worry about the virus, social distancing and the pandemics economic and social impact. QB was designed to provide a basic level of companionship during times of social distancing.

Ideally, QB functions as a close companion who is sensitive to the mental state of a user to respond to them accordingly. This was the principal idea behind the wearable stress monitor. By monitoring GSR, ST and HR, QB can "see" the state of its user by performing the stress detection calculation using these signals.

The physical companionship aspect was conceived after watching Disney's Big Hero 6. This was also partly inspired by the design philosophy of robots like PARO, the furry seal companion robot. In both the movie and with PARO, a physical robots presence aids its users in achieving a sense of human bonding. These ideas helped inspire the incorporation of the physical social robotic portion of QB.

In Big Hero 6, the robot, Baymax, is an artificially intelligent robot that performs all types of assistance and support to his human friend. This was the motivation for leveraging Amazon Web and Alexa Voice Services into the design. These platforms provide the necessary tools to offer voice interface and media interaction.

As discussed in Section II, isolation and worry over a looming virus has been linked to increases in stress. These increases directly correlate to increased negative health outcomes. Those living alone in isolation are at increased risk. Having as much access to physical companionship as possible is a vital component in mitigating these negative outcomes.

IX. Deployable Prototype Status

QB meets the feature specification of the original design idea contract specified in Section III with some modifications. As it stands, the design can detect phasic stress according to the stress detection algorithm based on GSR only. Wireless communication works as specified at a range encompassing most office or household settings. The interaction provided by QB to help relieve stress and provide companionship is difficult to quantify but qualitatively accomplishes what is specified in the design idea contract with quantitative methods available to support this conclusion.

A. Sensor Array

The sensor array performs the final function of sensing a candidate MOS and delivering a signal to the main QB robotic module indicating this. Currently, the array is deployed with HR, ST and GSR sensor equipment, however at the time of assembly, only the GSR and HR were functioning properly. Still, in the two of three real world tests, QB was able to sense a candidate MOS relying solely on GSR. This met the desired performance spec.

The GSR feature passes the required performance specification. Output voltage is amplified to levels differentiable by the 10-bit ADCs. In real world test scenarios, GSR samples are recorded and stored for the previous 20 seconds. The most recent sample is used in score calculation and is compared to the previous 20 seconds. According to literature, a deep breath triggers the opening of sweat glands, causing a rise in GSR. This was observed in every test case. *Figure 17* shows three drops in the waveform of about 250 mV each. These drops indicate peaking GSR. The magnitude of the drop is greater than the magnitude required to change a single big value of the 10-bit GSR (250 mV >> ~4 mV). These observations validated the performance of the GSR sensor and amplification circuitry.

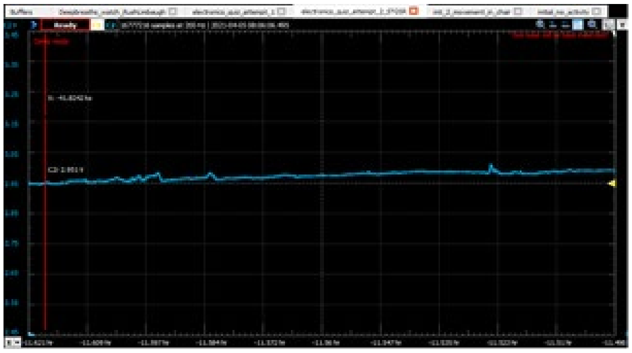


Figure 17. GSR during deep breathing [30]

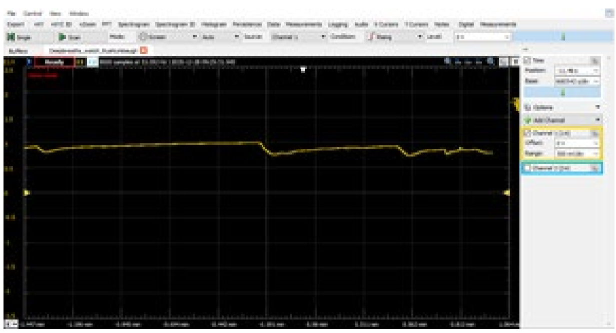


Figure 18. Voltage output of Skin Temp sensor [30]

Figure 18 shows the recorded waveform of ST during a user's Physics Exam. The resolution is 50 mV. The waveform shows differentiation of 30 mV. Other results show the temperature sensor to have a gain of about 1000 V/V and was capable of magnifying temperature variance of 0.1 degree C to values > ~4mV, which is required to change the LSB of the ADC by 1. This was accomplished with the amplifier built using the TL072 op-amp. The design was switched to the LM324 op-amp to accommodate the lower supply rail of the battery. Testing revealed the output was like before but had some inconsistencies. Because of time constraints, this build was implemented into the final wearable without further debugging. Final testing showed revealed that output was pulled to ground, instead of the op-amp's DC bias value. A potential cause of this is an undesired short somewhere at the input of the amplifier.

The HR feature performs as specified in the design idea contract. Observation of the output using the Arduino IDE's Serial Monitor demonstrated this. Verification of HR was done manually by counting a user's pulse.

The stress sensor still meets the criteria of detecting a potential MOS when worn by the user, but it's accuracy should be enhanced by debugging the ST amplification circuitry and tuning the rule weights in an iterative process.

B. Bluetooth

Implementation of Bluetooth according to the design idea contract was carried out as specified. Currently, two HC-05 breakout boards are connected to each of the microcontrollers in use for the wearable and stationary portion. These modules are paired with AT commands and communicate with the Serial Port Protocol: 1 start bit, 8 data bits, 1 stop bit and no parity bits. Final testing showed that input matched output in 10 out of 10 tests. The Baud rate for this module is 115200, the same as the programmed Baud rate of the Arduino and ESP-32 units.

C. Alexa Voice Services

The following test plan will focus in measuring the accuracy of Alexa Voice Services (AVS), Lambda Services, Cloud Database (DynamoDB), as well as the interaction between users and QB. With AVS enabled, the user can decide what QB can do to help the user in a moment of stress. The test plan created for this will focus on measuring responsiveness to the user's interactions. First, the team will test the accuracy of responsiveness by QB with the user's natural responses with the already created interaction model. The interaction model focus on 4 major categories to help someone relieve a moment of stress. These categories are Video, Music, Pictures and Exercises. The user will say their commands, and these will be used to either improve/change the current intents behind each category. By the end of this test, the team hopes to obtain at least 10 to 20 different utterances for each category. This test will help improve the interaction model to be more accurate and give a more expandable range of commands. Ideally, the team will also try to do this for at least one more language, such as Spanish, since QB will be supported in more than one language.

D. Lambda

For the Lambda Service the team will focus on the precision of recorded values for user's inputs. Lambda works as the backend piece of QB, and in order to make the responses that QB gives to the user are correct they must be tested. The test plan for this will focus on giving 5 commands and 5 false or a negative command. This will be repeated and used for each of the categories. Totaling at least 40 commands to measure accuracy of back-end interaction. the expected response of a negative/false command should be to adhere the user to ask for something else or ask the user to repeat the command. Since Lambda relies on AVS's speech-to-text model, it is not certain which words may cause QB to react inappropriately. All the values generated by the back-end platform will be submitted to the Cloud Database (DynamoDB). This offers another layer of accuracy for the team to measure since a comparison can be made between user-to-lambda, and lambda-to-dynamoDB. The test plan will generate an accuracy ratio between correct and incorrect commands between the 2 sets listed before.

E. Display/Webpage

The webpage will be tested to support a variety of formats for content delivery. The team has currently delivery of JPEG format but will be testing the usage of H.264, H.265, and HEVC codecs. Additionally, content in different resolutions such as 2160 @ 60hz, 2160 @ 24hz, 1440p @ 60hz, 1080p @ 60hz, 1080p @ 30hz, 720p @ 60hz, 720p @ 30hz, 480p @ 60hz, 480p @ 30hz. Additionally, the team will also test responsiveness and latency with the back-end model. Latency will be measured and recorded to provide a data point of delay between user's input and device's output.

F. Robotics & IoT

One major component of the product to be tested is the timing and syncing of devices to AWS. There are signals between the QB module and the AWS IoT shadow which is also changed by a Lambda function. We are going to view the state of IoT shadow as we test each prompt for the online side. For the QB module we will output the reading from the IoT shadow, and the decisions made to the console when testing QB's prompts. The physical actions from QB and all variables on console will be visible when testing. Next there is communication between QB and the wearable with sensors. Values must match between devices and trigger QB to update the IoT shadow appropriately. Again we will have to make use of the environment console to view variable data as the program runs. We will have to ensure that our protocols for data lost run appropriately and do not incorrectly trigger a stress detection. These three components, wearable, AWS, and QB, will have to run synchronously but this method ensures we can fully monitor the behavior of each.

G. Final Module

QB is designed to operate in an office environment. It is expected that temperature and humidity levels will remain consistent throughout the product's operating lifespan. Voltage and current regulation are handled by the development board in use for the prototype. Because of cost and time constraints, and the expected operating conditions of QB, testing each component with extreme voltage and current values was deemed unnecessary.

The fully assembled prototype has undergone a series of "real world" tests. These involved wearing the QB stress sensor while taking a physics exam. The exams lasted 15 minutes and the requirement to pass was the generation of a stress score with waveforms that matched what was predicted by the stress score algorithm: a peak in GSR with an associated temperature drop at the specified time interval. This was observed in 2 of the 3 tests. The reliability of Bluetooth communication and the appropriate responses provided by QB through physical, voice and media interaction demonstrate meeting over 90% of the specifications in the design idea contract.

X. MARKETABILITY FORECAST

A. Is the Interest/Demand There?

Pre-COVID, it was already commonly accepted that in general, day-to-day life was getting more and more stressful every passing year. A huge range of popular products ranging from things as simple as stress balls and fidget spinners, to essential oil diffusers and the explosion in popularity of CBD products all had things in common. They all have been marketed in some way to address some aspect of mental health, be it something as simple as helping one to focus, relax, or just distract themselves. They all seek to relieve some aspect, large or small, of the bigger picture of mental health. In more recent years, especially since COVID-19 has shown itself, there has been a boom in biomonitoring wearable devices with a focus on monitoring, managing and relieving stress, anxiety, depression, or any combination of the three. While QB was designed with the problem of stress in the forefront of our philosophy, anxiety and depression cannot simply be ignored as the three often tend to come as a package deal in some degree, large or small. On that same note, some statistics reported by CNBC state that between 2019 and 2020 the demand in the workplace for those with skill in stress management increased 1,015% and for those with anxiety management skills a formidable 3,867% [24]! It is particularly noteworthy that this time frame is before the global pandemic even asserted itself as a factor for our consideration. Clearly, even just a casual look at the market for stress reduction shows that the interest and demand is not only present, but it is healthy and growing. But let's take a more in depth look at the market landscape.

B. A Market Overlook

Without even looking at the hard data, our group was all pretty confident in the growth of the market for our product. Just looking at the global landscape and the recent boom in products with similar functionality (which we discuss later in this section). Sure enough, the market research does indeed confirm our assumptions. One marker research report published by BCC Research states, "The global stress management treatments market should reach $20.6 billion by 2024 from the $7.01 billion in 2019 at a compound annual growth rate (CAGR) of 3.7% for the forecast period of 2019 [25]." While this is not a huge rate of growth (for an already large market, 5-12% is generally considered a healthy CAGR), it is growth, nonetheless. A similar and more recent market research study reported by Globe News Wire states "The global workplace stress management market size is expected to reach $11.3 billion by 2025, rising at a market growth of 8.5% CAGR [26]." First, we must recognize the difference between these two reports. The first one is referring to the overall global market of stress management. The second report cited is specifically regarding stress management in the workplace. Secondly, the first was reported in 2019 before anyone had ever heard the word "coronavirus" before, the latter was reported in early 2020 when the coronavirus was somewhat known but had not yet hit its full force. One must assume that a savvy, competent researcher would indeed take this into account, however. While we cannot ignore the fact that both of these studies occurred before the pandemic hit its full swing, simple observation and deduction dictate that it is absolutely reasonable to conclude that the actual growth of this market has and will continue to surpass that which was projected in these two studies.

The social robotics market is projected to grow as well. Pre-pandemic studies show the social robot market was valued at $288.23 million in 2017 and is projected to expand at a compound annual growth rate of 13.75% to reach $699.18 million by 2023.

The target market segments are experiencing growth likely due to several drivers. These include increasingly disruptive work-life balance, decreasing stigma around mental disorders, recognition of the mental needs of adolescents, and the effects of COVID-19 and social distancing. The latter driver has caused a huge lack of social and physical interaction important to human mental health [27].

One specific example demonstrating the increased demand in stress relieving products due to social distancing is the increase in sale of automatic back massagers. "In the 6 months ending October 2020, 5.6 million massaging appliances were sold, an 86% increase compared to the same time last year. [...] The average amount spent on massaging appliances between January and October was 38% higher than last year, making it the category with the third largest increase in average selling price across all general merchandise industries. […] Dollar sales of massaging appliances continue to grow by more than 50% through [the 2020] holiday season [28].

C. Our Target Demographic

So just who is the target consumer for QB? QB is a virtual companion designed to assist in monitoring and reducing stress levels while attempting to alleviate the feeling of loneliness. Figures and statistics presented in earlier sections of this report have clearly established that the issue of stress as it relates to the ongoing pandemic knows no demographic. Across the board, regardless of all the different ways to differentiate a sample population, age, race, income, you name it, the problem exists. Not only that but the severity to which it affects any given demographics is comparable. With different countries in varying stages of their response to the pandemic there are still millions of people restricted to their homes for work and school. That is millions of people who may be going without their normal means of social support as also discussed earlier in this report. Even taking COVID out of consideration the problem of stress in society was already a formidable issue. However, COVID-19 is a reality we are unable to ignore. A collection of studies reported on by scientific news outlet The Conversation states that overall, mental health has declined during the pandemic. The statistics suggest that these declines are directly related to societal responses to the pandemic. "In Denmark, for example, mental health declined during the lockdown of the first wave but improved as the Danish government gradually reopened society[29]." The pandemic has only compounded the problem in a variety of ways. We conclude therefore that QB's target consumer can be just about anyone.

In the beginning phases of marketing, however, the initial target consumer for QB is parents dealing with children up to adolescence. QB provides them an opportunity to monitor and care for their wellbeing while being hands off. Research has shown that people may prefer opening up to robots as there is no bias or judgement involved. After establishing a presence with this initial target audience, the next targets would be clients needing to manage workplace stress, gamers, and then late adopters such as the elderly living alone or in nursing homes.

D. QB's Place in This Market

There are several products that work or perform similar functions that QB does. In the segment of mental health & fitness, devices like Whoop, Apple Watch, and Fitbit carry out daily trackers for heart rate, blood pressure, physical activity and heart rate variability. The data collected by these devices are then given to the user with the hope to improve tendencies and give advice for how the user can change for better. Some of the devices that execute personal assistance or companionship activities are Buddy, Kuri, Zenbo and Aido. All of these products perform actions such as responding to human touch, patrolling home while users are a way, facial recognition and offers a variety of fun and interactive games for children and adults. QB will be able to compete against these products since it offers similar features such as response to human touch, display functionality, vital tracking, voice interaction, and home connected capability since it powered via Alexa. QB's modular capability can also be a huge factor for improving its value in the market. Meaning that it can have more capabilities added via is voice design that will further enhance its value.

With regards to pricing, the breakdown of cost of materials is shown in Table 1. This puts cost of a single QB unit at *$160-$185*. Typically, wholesale price is the sum of the cost of labor and materials multiplied by 2 – 2.5 dictating a wholesale price of *$320 - $465*. Typical MSRB that a retailer will charge is another multiplication of 2 – 2.5 of the wholesale price. This puts QB’s retail price potentially at *$640 - $1165* which will bring it into the market around the upper-mid to high price range among its competitors. In comparison to other devices in this market we see prices ranging from the low hundreds to the thousands. For example, the Fitbit Watch, latest model, can cost $300, while a device like a companionship robot such as Misty from Misty Robotics can cost up to $3000.

Some companies like Whoop have arranged different payment plans in which the user can opt into a monthly plan of $30 and gains access to the device via this prepay subscription method. We could however also move for a “Buy Directly from the Manufacturer” business model to eliminate a vast portion of the retail markup and potentially offer QB for closer to the wholesale price, possibly a happy medium of $500. This would leave QB in the mid-range of prices among competitors with a formidable set of features compared to those competitors as well.

E. SWOT Analysis

A strength of the team is its small size and flexibility. This enables it to quickly adapt to the initial target market as well as early clients changing needs. Large OEMs would not have the dexterity to respond as quickly to a small market need. Another strength is the team's complimentary skill set. QB has many features requiring specialization in computer engineering, software development and electronics. One of the biggest strengths is in the teams understanding and experience in the problem being addressed. All members have personal experience dealing with the effects of extreme stress while being socially distanced.

Part	Cost (USD)
Controller Boards +	
Temp Sensor +	
Battery	$50
Amazon Echo Dot	$25
GSR Sensor	$10
Pulse Sensor	$25
Misc.	$50 - $75
Total	$160 - $185

Table 2. Cost Breakdown of a Single QB Unit

Weaknesses include the team's lack of expertise, funding and equipment and the modular work environment due in part to social distancing. This makes it difficult to integrate the features of QB. There is currently a lack of business acumen among team members.

Several opportunities are available to help drive QB’s success. These include social distancing, the $1.9 trillion-dollar economic relief package currently being debated by the Senate, the growing demand for social robots and the relatively cheap and accessible technology used to develop QB.

Threats to success include a flood of competitors. Many recent additions to the social robotics market accomplish similar tasks to QB, and so marketing QB’s different approach becomes a challenge. Also, the projected shortage of silicon manufacturing, current lack of OEM support, and social distancing restrictions may hamper production.

F. Manufacturing

Construction of version one of QB involves integrating various prefabricated products including the Arduino Lilly Pad, Echo Dot, three Adafruit sensors for GSR, temperature and heart rate and an HC-05 Bluetooth module.

The sensors will be soldered to a watch like casing along with the Arduino. Loops in the casing allow a watch strap to attach to a user’s wrist. The watch casing is fabricated with a 3D printer. The case design will need to be relatively large to allow room for all components.

The QB robotic module has several servo motors and a microcontroller that fit inside a custom 3D printed model. Components for arms, body and head are printed separately. The model is roughly 10 inches tall and can be printed with the same equipment used for the watch casing. Assembly involves simple hand tools to attach motor and limbs and an adhesive glue to cover these inner components with the printed body.

Depending on the level of demand, assembly will take place in the lab by the original designers. As demand increases to between 500 and 3000 units per month, assembly will be moved to leased manufacturing space within the US and involve several trained employees.

Supply of the wearable sensor's electronic components is subject to availability from outside manufacturers and vendors, and so a custom PCB that incorporates each of the necessary components is planned for the next QB version. The PCB design can be accomplished with any free design software and manufacturing can be accomplished overseas.

XI. CONCLUSION

This year's iteration of Computer/Electrical Engineering Senior Design at CSUS began in the midst of the COVID-19 pandemic. It is likely that every person on the planet has experienced its ramifications. Much of the world has resorted to social distancing in order to curb the spread of the virus, causing a tremendous number of unprecedented side-affects throughout society. Among the interconnected web of cause and effect is one problem that stood out to us as a team: increase in stress. Not only is the global climate exponentially more and more stressful every day due to the pandemic itself and societies response, but access to support for those who suffer from it has diminished as well. This is the societal problem we as a team decided to address in Senior Design. The way we decided to help alleviate this problem was with a companion robot to simulate the social support lost during social distancing and isolation from quarantine. The project was dubbed Quarantine Buddy, or QB for short.

The design of QB will help those in need when ever increasing moments of stress arise by acting as a personal companion. One of the key features of QB is the ability to understand, record, and obtain valuable data metrics about its user. These metrics are skin temperature change, galvanic skin response and heart rate. The sensors are designed to be controlled by a single microcontroller and housed in a device resembling a wearable watch. The sensors that will be used for capturing this data will communicate with an Arduino board via Bluetooth. In order to create a more "life like" experience, the team designed QB to implement servo motors that allow its head and arms to perform specific movement. These gestures will serve as physical feedback to the user after performing specific actions or in reply to a question from QB. Our design will take advantage of some of the cloud computing servers that are currently available in the market. Amazon Web Service (AWS) cloud computing will be leveraged to implement multiple functions as a cloud storage unit, and for voice activation. Using software such as Alexa Voice Services will allow the team to utilize the already existing architecture from Amazon to create a voice interface in which user can communicate with QB. While many products are available that serve a similar purpose, this prototype demonstrates a unique blend of features for tackling increased stress in a lonely environment.

The work breakdown structure of the project served as an aid for us to directly observe all the different tasks, the sum of which will equal a fully functional and fully realized project. Due to the requirement of a modular design, necessary in this COVID environment, we were able to see the workflow of all the team members laid out together and reevaluate some of the tasks before us. We were forced to reorganize some of the workflow to relieve the pressure from some of the other team members. The project timeline portion of the breakdown structure also put into perspective the time we have left and how best to utilize it. Through our risk analysis we were able to lay out the crucial paths of our project and anticipate potential problems that might hinder its success. Some of these anticipated issues were a direct result of the pandemic itself. The social distancing dictated not only by common sense but also as mandatory by rules of the senior design class led to a tumultuous development process. Differences in schedules and the difficulties of remote learning have been compounded by the realities of the pandemic. The risk assessment serves to act as a contingency plan for such events and is used to navigate us through the semester in times of unexpected, but not unprepared for, difficulties.

The basic design philosophy of QB is to develop a physically interactive companion who can “see” when someone needs help. The design needed to be executed by individuals who are themselves quarantining and aims to mitigate the negative consequences of a socially distanced future. There is room for improvement of the current feature set as well as opportunity for additional applications, but in its current state, this concept has been delivered with the current prototype.

REFERENCES

[1] Panchal, N., Kamal, R., Orgera, K. and Cox, C., “The Implications of COVID-19 for Mental Health and Substance Use,” *Kaiser Family Foundation,* Aug. 21 2020. [Online]. Available: kff.org/coronavirus-covid-19/issue-brief/the-implications-of-covid-19-for-mental-health-and-substance-use.

[2] United States Department of Labor., Civilian Unemployment Rate., Washington DC: United States Bureau of Labor Statistics, 2020. [Online]. Available: bls.gov/charts/employment-situation/civilian-unemployment-rate.htm.

[3] National Center for Biotechnology Information, “Effects of Unemployment on Mental and Physical Health.” *National Center for Biotechnology Information*, May 1985. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1646287/

[4] Holingue, C., Badillo-Goicoechea, E., Riehm, K., Veldhuis, C., Thrul, J., Johnson, R., Fallin, M., Kreuter, F., Stuart, E. and Kalb, L., 2020. *Mental Distress During The COVID-19 Pandemic Among US Adults Without A Pre-Existing Mental Health Condition: Findings From American Trend Panel Survey*. [online] Science Direct. Available at: <https://www.sciencedirect.com/science/article/pii/S0091743520302553?via%3Dihub> [Accessed 13 September 2020].

[5] United Nations Educational, Scientific and Cultural Organization., *Global monitoring of school closures caused by COVID-19.,* UNESCO Institute for Statistics, 2020. [Online]. Available: en.unesco.org/covid19/educationresponse.

[6] American Psychological Association, “Stress in America: Stress and Current Events. Stress in America Survey,” *American Psychological Association,* 2019. [Online]. Available: apa.org/news/press/releases/stress/2019/stress-america-2019.pdf

[7] Centers for Disease Control and Prevention, “Coping with Stress,” *Centers for Disease Control and Prevention,* July 1, 2020.[Online]. Available: cdc.gov/coronavirus/2019-ncov/daily-life-coping/managing-stress-anxiety.html.

[8] Ann Pietrangelo, “Suicidal Ideation is Increasing During COVID-19: How Best to Cope,” H*ealthline Media,* August 2020. [Online] Available: https://www.healthline.com/health-news/suicidal-ideation-is-increasing-during- covid-19-how-best-to-cope#Why-some- groups-are-more-affected [Accessed Sept. 25, 2020].

[9] The American Institute of Stress, “Stress Effects,” *The American Institute of Stress,* 2020. [Online]. Available: https://www.stress.org/stress- effects [Accessed Sept. 25 2020].

[10] K. Kyriakou, B. Resch, G. Sagl, “Detecting Moments of Stress from Measurements of Wearable Physiological Sensors,” *National Center for Biotechnology Information.* Sept. 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6749249/

[11] Arsalan, A., Majid, M., Rauf Butt, A., Muhammad Anwar, S., "Classification of Perceived Mental Stress Using A Commercially Available EEG Headband," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, issue 6, pp. 2257- 2264. [Online]. Available: ieeexplore.ieee.org/document/8753565.

[12] Y. Can, N Chalabionloo, D. Ekiz and C Ersoy, "Continuous Stress Detection Using Wearable Sensors in Real Life: Algorithmic Programming Contest Case Study," *Multidisciplinary Digital Publishing Institute*. 2019. [Online]. Available: https://www.mdpi.com/1424- 8220/19/8/1849. [Accessed Sept. 25, 2020]

[13] Alpha-Stim, "Alpha Stim technology: Fast,safe relief from pain, anxiety, depression, & insomnia," *Electromedical Products International, Inc.* [Online]. Available: alpha- stim.com/about-epi/.

[14] X. Ma, Z. Yue, Z. Gong, "The Effects of Diaphragmatic Breathing on Attention, Negative Affect and Stress in Healthy Adults," *National Center for Biotechnology Information*, Jun. 2016 [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5455070/ [Accessed Sept. 26 2020]

[15] A. Prestin, R. Nabi, "Media Prescriptions: Exploring the Theraputic Effects of Entertainment Media on Stress Relief, Illness Symptoms, and Goal Attainment," *International Communication Association.* April 2020. [Online] Available: https://academic.oup.com/joc/articl e/70/2/145/5821497 [Accessed Sept 26, 2020]

[16] National Institute of Mental Health, "Supporting Mental Health During the COVID-19 Pandemic," *National Institute of Mental Health,* April 2020. [Online] Available: https://www.nimh.nih.gov/news/science-news/2020/supporting-mental-health- during-the-covid-19-pandemic.shtml

[17] Lopes, L., Kearney, A., Kirzinger, A., Hamel, L., Brodie, M., "KFF Health Tracking Poll – December 2020: COVID-19 and Biden's Health Care Agenda," *Kaiser Family Foundation,* Dec 18, 2020. [Online.] Available: https://www.kff.org/coronavirus-covid-19/report/kff-health-tracking-poll-december-2020/

[18] Jabri A, Kalra A, Kumar A, et al. Incidence of Stress Cardiomyopathy During the Coronavirus Disease 2019 Pandemic. JAMA Netw Open. 2020;3(7):e2014780. doi:10.1001/jamanetworkopen.2020.14780

[19] Ettman CK, Abdalla SM, Cohen GH, Sampson L, Vivier PM, Galea S. Prevalence of Depression Symptoms in US Adults Before and During the COVID-19 Pandemic. JAMA Netw Open. 2020;3(9):e2019686. doi:10.1001/jamanetworkopen.2020.19686

[20] Lakhan, R., Agrawal, A., Sharma, M., "Prevalence of Depression, Anxiety, and Stress during COVID-19 Pandemic," *Journal of Neurosciences in Rural Practice,* Sept 11, 2020. [Online.] Available: https://www.thieme-connect.com/products/ejournals/abstract/10.1055/s-0040-1716442

[21] "Fitbit Official Site for Activity Trackers & More," Fitbit, Inc. [Online]. Available: fitbit.com/global/us/home

[22] Chaturvedi, A., “Robotic companion to help astronauts tackle stress and loneliness in space,” *Geospatial World*, July 26, 2018. [Online]. Available: geospatialworld.net/blogs/robotic-seal-astronauts-tackle-stress/

[23] Amazon. “AWS Lambda FAQs,” amazon.com. [Online]. Available: aws.amazon.com/lambda/faqs/

[24] Liu, J. “Training for This Workplace Skill Is Up Nearly 4,000% in 2020,” *CNBC*, Nov 30, 2020. [Online.] Available: https://www.cnbc.com/2020/11/27/training-for-this-workplace-skill-is-up-nearly-4000percent-in-2020.html?cm_mmc=membership-_-email-_-undefined-_-MBR-BuzzNewsletter-Email-MembershipBenefits-1228Buzz-19559&utm_source=membership&utm_medium=email&utm_campaign=MBR-BuzzNewsletter-Email-MembershipBenefits-1228Buzz-19559

[25] Javed, M. “Stress Management Industry: Global Trends,” *BCC Research.* March 2020. [Online.] Available: https://www.bccresearch.com/market-research/healthcare/stress-management-industry-report.html#:~:text=The%20global%20stress%20management%20treatments,period%20of%202019%20to%202024

[26] *GlobeNewsWire.* “The Global Workplace Stress Management Market size is expected to reach $11.3 billion by 2025, rising at a market growth of 8.5% CAGR during the forecast period,” Feb 13, 2020. [Online.] Available: https://www.globenewswire.com/news-release/2020/02/14/1985029/0/en/The-Global-Workplace-Stress-Management-Market-size-is-expected-to-reach-11-3-billion-by-2025-rising-at-a-market-growth-of-8-5-CAGR-during-the-forecast-period.html

[27] Transparency Market Research, “Management/Relief Supplements Market - Global Industry Analysis, Size, Share, Growth, Trends and Forecast 2019 – 2027," 2019. [Online] Available: https://www.transparencymarketresearch.com/stress-management-relief-supplements-market.html

[28] Marshall, Janine, “Consumer Purchases Reveal Desire for Stress Relief and Self-Care While at Home, Reports NPD,” Dec 23, 2020. [Online] Available: https://www.npd.com/wps/portal/npd/us/news/press-releases/2020/consumer-purchases-reveal-desire-for-stress-relief-and-self-care-while-at-home--reports-npd/

[29] Santini, Z.I., Koyanagi, A., “Feeling Lonely During the Panemic? You’re Up to Ten Times More Likely to Have Worse Mental Health,” Jan 27, 2021. [Online.] Available: https://theconversation.com/feeling-lonely-during-the-pandemic-youre-up-to-ten-times-more-likely-to-have-worse-mental-health-153064

[30] Lugo, V., Rodriguez, A., Tapia, Z., Thorpe, J.

[31] Mouser.com, “ZTP-115M Termophile IR Module,” 2020. [Online]. Available: https://www.mouser.com/catalog/specsheets/amphenol_08112017_AAS-920-550A-ZPT-115M-032114.pdf [Accessed May 1, 2020].

[31] seeedstudio.com, “Grove-GSR Sensor,” 2020. [Online]. Available: https://wiki.seeedstudio.com/Grove-GSR_Sensor/ [Accessed May 1, 2020].

GLOSSARY

Amazon Alexa – Or just '***Alexa'***. a virtual assistant AI technology developed by Amazon.

Amazon Simple Storage Service (S3) - A service through AWS that provides object storage through a web service interface.

Alexa Voice Service (AVS) – a cloud-based service that provides APIs to interface with Alexa.

Amazon Web Services (AWS) – a service providing on-demand cloud computing platforms and APIs to individuals, companies, and governments on a metered pay-as-you-go basis.

American Institute of Stress (AIS) - a non-profit organization which imparts information on stress reduction, stress in the workplace, stress related to military service and the health consequences of chronic, unmanaged stress.

API – Application Programming Interface. A computing interface that defines interactions between multiple software intermediaries.

Centers for Disease Control and Prevention (CDC) - a national public health institute in the United States. It is a United States federal agency, under the Department of Health and Human Services.

COVID-19 - an infectious disease caused by severe acute respiratory syndrome coronavirus 2. It was first identified in December 2019 and has resulted in an ongoing global pandemic. Also, colloquially, though inaccurately, interchangeable with "coronavirus".

CSS – *Cascading Style Sheets.* Style sheet language used for presenting documents written in HTML. This is what gives webpages their unique aesthetic flavor.

Electroencephalography (EEG) - a test that detects electrical activity in your brain using electrodes attached to the scalp. It is used to detect abnormalities in brain waves and in brain electrical activity.

Fuzzy logic - a form of many-valued logic in which the truth tables for a variable may be any real number between 0 or 1 both inclusive. Used to handle the concept of "partial truth", or more practically to apply a weight to a value within a given dataset. This contrasts with the far more commonly used Boolean logic, where a variable may only hold the value 0 or 1.

Galvanic skin response (GSR) - the property of the human body that causes continuous variation in the electrical characteristics of the skin. Varies with perspiration, psychological or physiological arousal. Observed as a measure of emotional and sympathetic responses. Also known as *skin conductance, electrodermal activity/response, etc.*

GUI – *Graphical User Interface.* The interface allowing users to interact with electronic devices or software.

Heart rate variability (HRV) - the variation in the time interval between heartbeats. *Related to RR variability.*

HF – high frequency.

HTML – *Hypertext Markup Language.* The standard markup language for documents designed to be displayed in a web browser.

International Communication Association (ICA) - an academic association for scholars interested in the study, teaching and application of all aspects of human and mediated communication.

JS – *JavaScript.* The programming language that enables interactive webpages.

k-Nearest Neighbor (kNN) - a method of pattern recognition used for classification and regression.

Kaiser Family Foundation (KFF) - a non-profit organization focusing on major health care issues facing the United States, as well as the country's role in global health policy. A non-partisan source of facts and analysis, polling and journalism for policymakers, the media, the healthcare community, and the general public. Formerly known as *The Henry J. Kaiser Family Foundation.*

Linear discriminant analysis (LDA) - a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. Also known as *normal discriminant analysis (NDA)* or *discriminant function analysis.*

LF – low frequency.

Machine learning (ML) - the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence (AI). Machine learning algorithms build a mathematical model from sample data or "training data", to make predictions and/or decisions they haven't been explicitly programmed to.

MOS – moment of stress.

National Center for Biotechnology Information (NCBI) - Houses a series of databases relevant to biotechnology and biomedicine and is an important resource for bioinformatics tools and services. Part of the *United States National Library of Medicine* (NLM), itself a branch of the *National Institutes of Health* (NIH).

National Institutes of Health (NIH) - the primary agency of the United States government responsible for biomedical and public health research.

National Institute of Mental Health (NIMH) - The largest research organization in the world specializing in mental illness. One of 27 institutes and centers that make up the *National Institutes of Health* (NIH), which itself is an agency of the *United States Department of Health and Human Services.*

RR-interval – the interval between consecutive peaks, RR, on a wave usually representing heart rate. Used to calculate instantaneous heart rate. *See Heart rate variability.*

S3 *– see Amazon Simple Storage Service.*

Support-vector machine (SVM) - supervised learning models in associated with learning algorithms that analyze data used for classification and regression analysis. Also known as *support-vector network.* Part of *machine learning (ML).*

United Nations Educational, Scientific and Cultural Organization (UNESCO) - a specialized agency aimed at promoting world peace and security through international cooperation in education, the sciences, and culture.

United States National Library of Medicine (NLM) - the world's largest medical library. Operated by the United States federal government. It is an institute within the National Institutes of Health (NIH).

User Manual:

The Quarantine Buddy (QB) is an interactive robot that is meant to be kept at home near a computer or tablet. At times where you will be indoors for longer than usual or when you are feeling more anxious of going outdoors, power on and put on the stress detection wearable. QB and the Wearable have Bluetooth functionality so feel free to continue with your normal activities. If a moment of stress is detected, QB's eyes will light up. If you notice QB's eyes are on it is recommended you interact with QB. When doing so, have your device on with the QB webpage open in your web browser. You can also interact with QB at any time with the wearable on or off. QB will respond differently based on whether stress was detected. To communicate with QB, press the speech button on you echo dot and say "hey QB" or ask Alexa to "open hey QB."

Setup Requirements:

- Amazon account
- Download the Alexa Skill

Powering and Using it the Wearable:

1. Strap the glove to the hand with the IRT sensor placed above the center of the back of the hand.
2. Place one lead on the same hands index finger and one lead on the middle finger.
3. Turn on the switch at the center of the control board to the "ON" state.
4. If LED on the HR monitor, or Arduino board does not light when power switch is in the "ON" state, the wearable battery may need recharging.
5. When running, Skin Temperature and Galvanic Skin Response will be used to determine your stress response every second. If a stressful state is determined, QB will know.
6. When a stress response is detected, QB's eyes will light up. QB's eyes are on when he senses you are stressed. When you notice this, you should talk to QB and let him know you're ok or that you could use some help.
7. To talk to QB, press the button on the Echo dot and say, "Hey Q-B."
8. Follow QB's responses. When talking with QB, you will notice QB perform certain gestures and the Webpage change in appearance to display videos or videos.
9. Powering QB: QB itself plugs directly into the wall outlet of your home. The sensor hand-strap is powered by a built-in rechargeable 3.7V Lithium-Ion battery.

QB requires internet access to fully function. A microUSB power source is required for each the ESP32 and the Echo Dot. Last a 5V power source is required for powering the Servo motors.

When talking with QB, you will notice QB perform certain gestures and the Webpage change in appearance to display videos or videos.

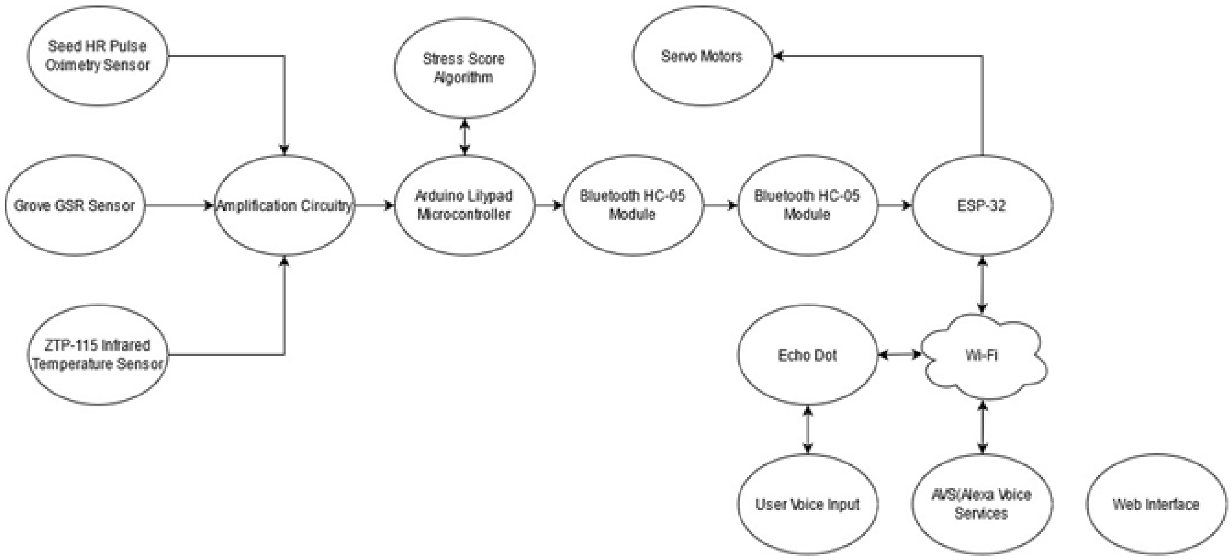


Figure B-1. Hardware layout of QB [30]

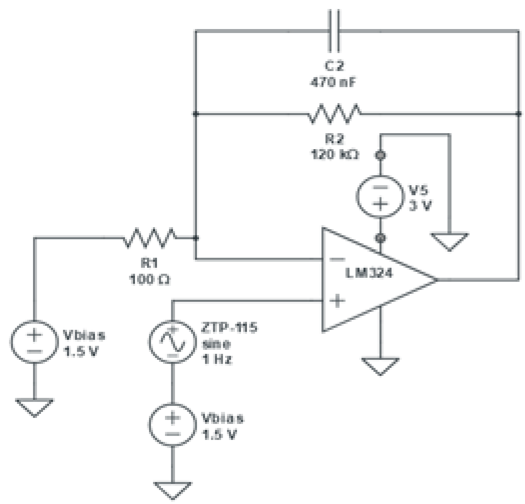


Figure B-2. Skin Temp Amplifier [30]

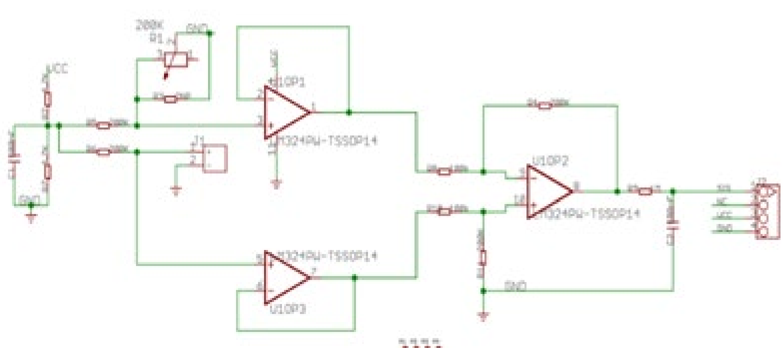


Figure B-3. GSR Amplifier [32]

Feature to be Tested	Components	Approach	Pass/Fail Criteria	Test Deliverables
Sensor Array				
	GSR			
		Oscilloscope used to measure reading during "Stroop test"	GSR rises in 7/10 stress events	GSR reliability
		Arduino IDE or Quartus software used to verify ADC is differentiable for Max/Min Voltage readings	GSR rises in 7/10 stress events	GSR Data for Stress-score calculation
		Compare readings to commercially available sensor	readings fall within 10% the corresponding value of the other sensor 10 tests in a row	GSR accuracy
	ST			
		Oscilloscope used to test Max/Min Voltage Swing	Pass- Voltage Swing of 1V under normal user skin temperatures	User ST range
		Arduino IDE or Quartus software used to verify ADC is differentiable for Max/Min Voltage readings	Binary Output resolution of > 10 bits over defined temperature ranges	ST data for Stress-score calculation
		Compare readings to commercially available sensor	readings fall within 10% the corresponding value of the other sensor 10 tests in a row	ST accuracy
	HR			
		Compare readings to commercially available sensor	readings fall within 10% the corresponding value of the other sensor 10 tests in a row	HR accuracy
	Bluetooth Module			
		Arduino IDE software to generate and record signals	In 5/5 tests, and in each test, 100 integer values are being sent, 95% of signals are recorded accurately at a range of 20m	Wireless capability
Final Prototype User Test				
		Using Online Stroop test, users are given 10 minutes to pass a series of tests.	QB stress monitor generates score >= threshold (150) in 3/4 tests.	Stress Algorithm Validation
		Users take psycho-analytic stress tests completed 15 minutes before and immediately after stress tests. Stress signal recordings are used to determine correlation between signal readings before, during and after stress events.	Stress readings that correlate with reduced stress after stress test and QB interactions match with user PA tests in 2/3 subjects who generated a threshold test score.	Prototype stress relief validation.
Device AWS IoT				
	Shadow Reads and Updates			
		Monitor shadow variables as prompts answered for correct updates and responses from Alexa	Alexa Follows correct conversation flow 80% correct out of 10 tests	QB IoT and Lambda Function Communication Functionality
		Monitor shadow variable as QB Module determines stress and provides update	IoT shadow is updated appropriately 80% correct out of 10 times when stress detected	QB Module Online Communication Functionality
Robotic Limbs				
	Servo Motors			
		Correct motors triggered with intended gestures during conversation queues with QB	4 out of 5 test, correct gestures triggered during the five complete conversations with QB	QB Robot Companionship Presence

Table B-1 Device Testing Plan / Results

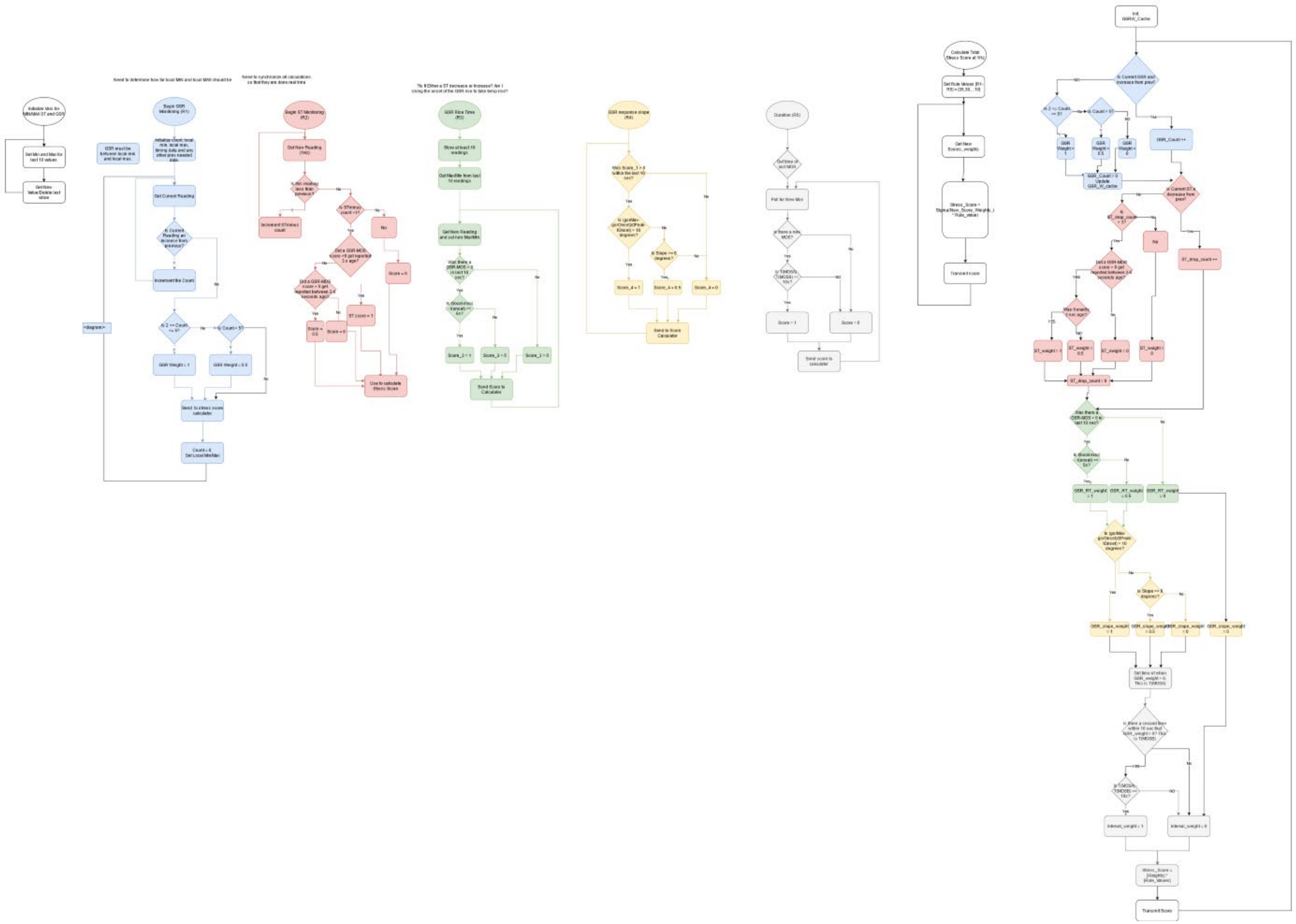


Figure C-1. Logic Flowchart for Stress Detection Algorithm

```
#include <SoftwareSerial.h>

/*****************/
/*VERSION 1.00.00*/
/*****************/
#define USE_ARDUINO_INTERRUPTS true
#include "sensors.h"
#include <math.h>


PulseSensorPlayground pulseSensor;
OneWire oneWire(TEMP_PIN);
DallasTemperature sensors(&oneWire);

float degC, degF;                           //var to hold temp measurements
int test_count = 0;
int BPM;                                    //var to hold BPM
int GSR, GSRBase, GSRVal, stressScore;      //vars for setting up GSR and hold measurement
float gsrSlope;
bool change;                                //if emotional change detected
int gsrSample[SAMPLE_SIZE];                 //array to store GSR readings
int gsr_onset_t[SAMPLE_SIZE];              //stores onset based on rise time of GSR
int gsr_onset_value[2*SAMPLE_SIZE];
float tempSample[SAMPLE_SIZE];              //array to store temp readings
int scoreSample[SAMPLE_SIZE];               //array to store stress score readings
int gsrOnset[SAMPLE_SIZE] = {0};            //stores R0 weights for last 10 sec.//initiate as 0
bool gsrFilled = false, tempFilled = false, scoreFilled = false; //to indicate when arrays are initially filled
int gsrCount = SAMPLE_SIZE - 1;             //used for initial filling of arrays
int tempCount = SAMPLE_SIZE - 1;            //used for initial filling of arrays
int scoreCount = SAMPLE_SIZE - 1;           //used for initial filling of arrays
int gsrMax, gsrMin;
float tempMax, tempMin, gsrAvg, tempAvg;
int gsr_rise_cnt = 0;                         //used for stress score
int st_fall_cnt = 0;                          //used for stress score
int rule[5] = {0};                          //holds rule scores, 0, 1, 2. change to byte for space
static int weight[5] = {30, 25, 20, 15, 10};//holds the rule weights
int threshold = 75;

void setup() {
  long sum = 0;
  Serial.begin(115200);

  /*Set up pulse sensor*/
  pulseSensor.analogInput(PULSE_PIN);
  pulseSensor.setThreshold(HR_THRESH);
  pulseSensor.begin();
  /*********************/

  /****************************/

  /*Starts the temp sensor*/
  //sensors.begin();
}

void loop()
{
  sensors.requestTemperatures();
  //degC = sensors.getTempCByIndex(0);
  //degF = sensors.getTempFByIndex(0);
  degF = analogRead(TEMP_PIN);
  BPM = pulseSensor.getBeatsPerMinute();
  GSR = getGSR();
  //change = abs(GSRBase - GSR) > GSR_THRESH; //if GSR deviates from baseline

  if (!gsrFilled) {                           //if array not yet filled
    gsrSample[gsrCount] = GSR;                //initialize array starting with highest index going down
    gsrCount--;
    if (gsrCount < 0)                         //if lowest index has been filled
      gsrFilled = true;                       //array is full
```

```
  } else {
    addToSample(gsrSample, SAMPLE_SIZE, GSR);
                                       gsrMax = getMin(gsrSample, SAMPLE_SIZE);//higher GSR has a lower reading
    gsrMin = getMax(gsrSample, SAMPLE_SIZE);//lower GSR has a higher reading
    gsrAvg = getAvg(gsrSample, SAMPLE_SIZE);
  }

  if (!tempFilled) {                          //if array not yet filled
    tempSample[tempCount] = degF;             //initialize array starting with highest index going down
    tempCount--;
    if (tempCount < 0)                        //if lowest index has been filled
      tempFilled = true;                      //array is full
  } else {
    addToSample(tempSample, SAMPLE_SIZE, degF);
    tempMax = getMax(tempSample, SAMPLE_SIZE);
    tempMin = getMin(tempSample, SAMPLE_SIZE);
    tempAvg = getAvg(tempSample, SAMPLE_SIZE);
  }

  //==================================RULE 0: Is current GSR reading an INCREASE? ================================
  //===============if it increases between 2-5 sec, max score, else increases more than 5 sec, half score==========
  //int gsrTemp = 0;
  if(gsrSample[0]<gsrSample[1]){//is most recent an increase?
    gsr_rise_cnt ++;//
    rule[0] = 0;//
    addToSample(gsr_onset_t, SAMPLE_SIZE, 0);
    addToSample(gsr_onset_value, 2*SAMPLE_SIZE, GSR);
  } else {// check count and give Rule_1 score
    if(gsr_rise_cnt <= 5 and gsr_rise_cnt >= 2){
      rule[0] = 2;//MAX GSR score
      addToSample(gsr_onset_t, SAMPLE_SIZE, (gsr_rise_cnt + 1));//track the onset time, onset = time before GSR
began to rise
      addToSample(gsr_onset_value, 2*SAMPLE_SIZE, GSR);//use for gsr slope calculation
    } else {
      if(gsr_rise_cnt > 5){
        rule[0] = 1;//half GSR score
        addToSample(gsr_onset_t, SAMPLE_SIZE, (gsr_rise_cnt + 1));//
        addToSample(gsr_onset_value, 2*SAMPLE_SIZE, GSR);//
      } else {
        rule[0] = 0;
        addToSample(gsr_onset_t, SAMPLE_SIZE, 0);//
        addToSample(gsr_onset_value, 2*SAMPLE_SIZE, GSR);//
      }
    }

    gsr_rise_cnt = 0;//restart gsr rise count
  }


  //======================RULE 1: Is current ST Reading a DECREASE? [Tt+3:Tt+m]’ < 0 ==========================//
  //======Tt is the time of stress onset where GSR begins to increase. Tt+3 is 3 seconds after this onset=======//
  int x = 0;
  int y = 0;
  int MOStime = -1;
  rule[1] = 0;
  for(y = 0; y < SAMPLE_SIZE; y++){
    if(rule[1] > 0){
      break;
    }
    if(gsr_onset_t[y] > 0){
      MOStime = y+gsr_onset_t[y];
    }//first if
    if(MOStime > -1 and MOStime <= SAMPLE_SIZE + 3){//stay within temp array boundaries
      int STinitial = tempSample[MOStime -3];
      //compare to now
      if(STinitial > tempSample[0]){
        if((MOStime - 3) > 3){
          rule[1] = 2;
```

```
          //should also update rule[0] to the rule[0] weight at MOStime = x + gsr_onset_t[x]? review documentation
1st.
        }
      }//compare STt+3
      if(rule[1] != 2 and rule[1] != 1){
        for(x = 2; x <=6; x++){
          if((MOStime - x) >= 0){
            if(tempSample[MOStime - x] > tempSample[0]){
              if((MOStime - 3) > 3){
                rule[1] = 1;
                //should also update rule[0] the rule[0] weight at MOStime = x + gsr_onset_t[x]? review
documentation
              }
            }
          }
        }
      }//compare STt+(2-6) to ST now
      if(rule[1] != 2 and rule[1] != 1){
        rule[1] = 0;
      }
    }
  }//first for

  //Pseudo code
    //if you find one, record MOStime = k + gsr_onset_t[x]
    //if MOStime > 0
    //get ST val at Tt+3
      //compare STval(Tt+3) to now
      //is result < 0?
      //what is the time difference?
      // >= 3 seconds?
      // rule[1] = 2
      //
    //if this returned nothing, do the same at Tt+2, Tt+4, Tt+5 & Tt+6

  //====================================RULE 2: GSR rise time ==================================================
//loop through gsrSample[k] to find the index of the max
  int i = 0;
  int max_t = 0;
  int gsr_start_time = -1;
  int gsr_max_t = -1;
  int gsr_rise_t = -1;
  //Serial.print("GSR max time: ");
  //Serial.println(max_t);
//loop through gsr_onset_t to find the start time
  for(i = 0; i<SAMPLE_SIZE; i++){
    if(gsr_onset_t[i] > 0){
      gsr_start_time = (i + gsr_onset_t[i]);
      break;
    }
  }
  //loop through and find the value of the most recent gsr max
  for(i = 0; i<SAMPLE_SIZE; i++){
    if(gsrSample[i] == gsrMax){
      gsr_max_t = i;
      break;
    }
  }
  if(gsr_max_t != -1 and gsr_start_time != -1){
    gsr_rise_t = gsr_start_time - gsr_max_t;
  } else {
    gsr_rise_t = 20;
  }
  if(gsr_rise_t <= 5){
    rule[2] = 2;
  } else {
    if(5 < gsr_rise_t and gsr_rise_t <= 15){
      rule[2] = 1;
    } else {
```

```
      rule[2] = 0;
    }
  }

    //====================RULE 3: GSR slope. (GSRpeak-GSRonset)/peak_t-onset_t =================================
//  get most recent onset time value:
  float num = -1;
  float den = -1;
  int k;
  int l;
  for(k = 0; k<SAMPLE_SIZE; k++){
    if(gsr_onset_t[k] > 0){
      num = abs(gsrMax - gsr_onset_value[k + gsr_onset_t[k]]);
      break;//using most recent MOS candidate
    }
  }
  //get time of Peak and set den
  for(l = 0; l<SAMPLE_SIZE; l++){
    if(gsrMax == gsrSample[l]){
      den = abs(l - (k+ gsr_onset_t[k]));
    }
  }

  if(num >= 0 and den >= 0){//checks to see if there was an onset and if the slope rose
    gsrSlope = (atan(num/den))*(180/(3.14));
  } else {gsrSlope = 0;}
  if(gsrSlope >= 10){
    rule[3] = 2;
  } else if(8 <= gsrSlope and 10 >= gsrSlope){
    rule[3] = 1;
  } else{
    rule[3] = 0;
  }


//==========================RULE 4: check for multiple onsets within last 10 sec ================================
  int onset_cnt = 0;
  for(k = 0; k < SAMPLE_SIZE; k++){
    if(gsr_onset_t[k] > 0){
      onset_cnt = onset_cnt ++;
    }
  }
  if(onset_cnt > 1){
    rule[4] = 2;
  } else {
    rule[4] = 0;
  }
//=======================================END MOS Detection Rules ================================================

//================================Calculate Final Stress Score====================================================
stressScore = 0;
for(k = 0; k < 5; k++){
  stressScore = stressScore + (rule[k] * weight[k]);
}

int data;
if(stressScore >= threshold){
  data = 1;
} else {
  data = 0;
}
Serial.write(data);
  stressScore = 0;

delay(700);                      //sample every (INTERVAL / 1000) seconds
}
```

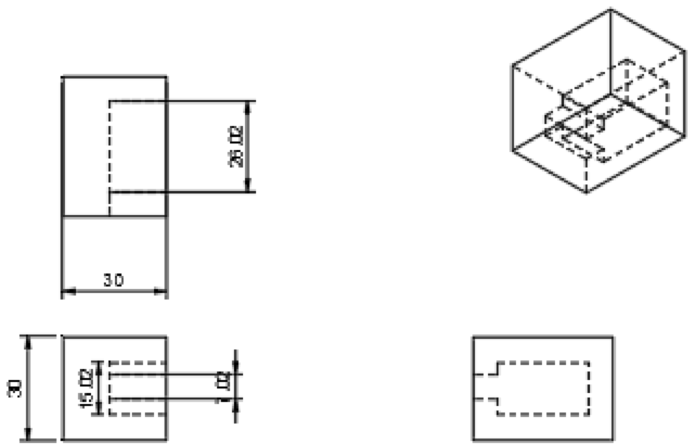


Figure D-1. Elbow Model [30]

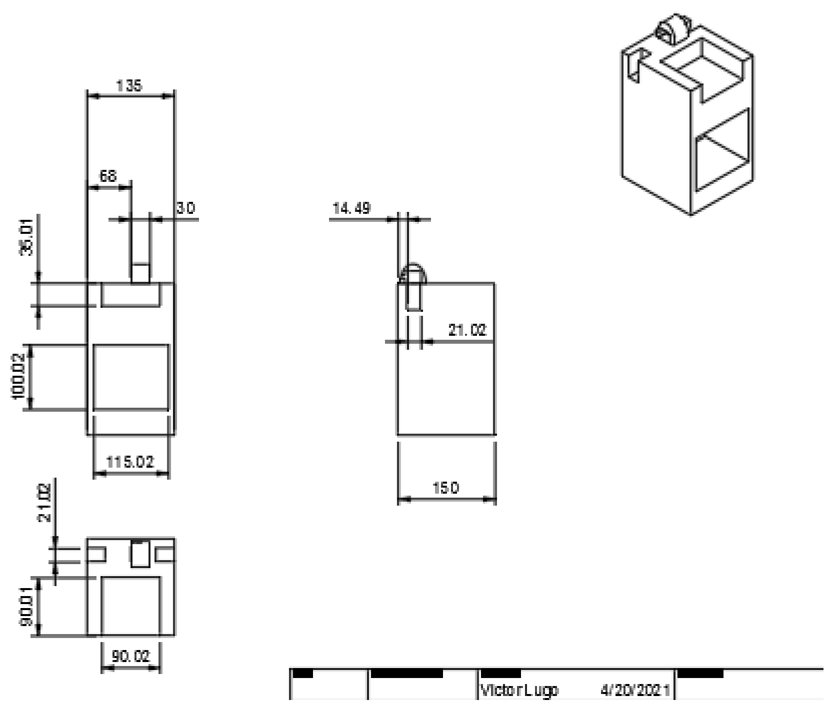


Figure D-2. Frame Model [30]

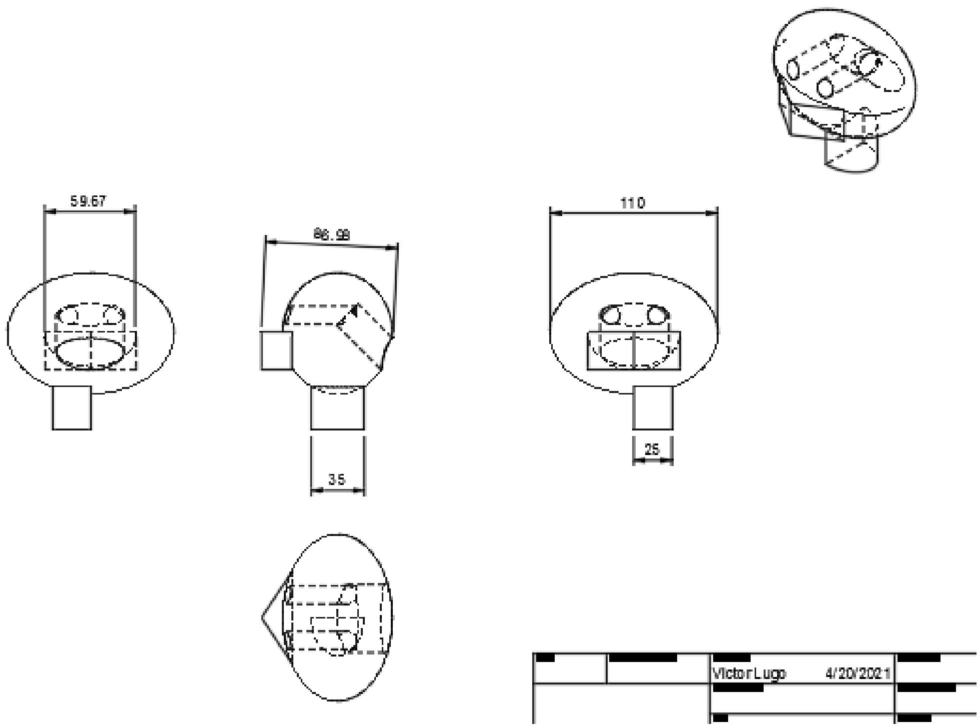


Figure D-3. Head Model [30]

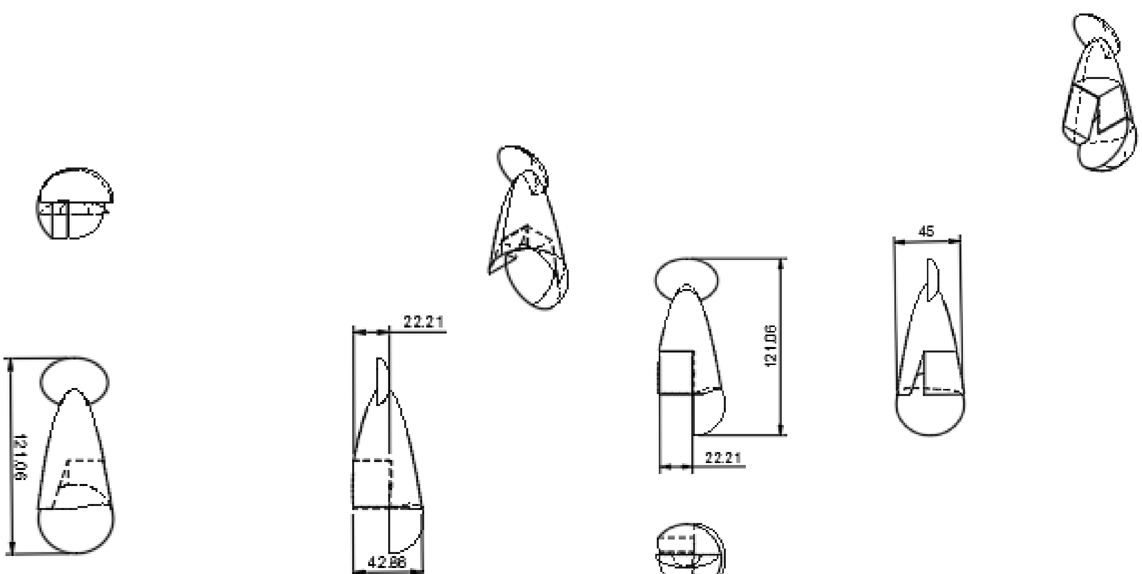


Figure D-4. Left arm model (Left) & Right arm model (Right) [30]

Professor Neal Levine, CSUS – Main point of contact throughout the project for support, advice, insight, clarification, project specifications, requirements, etc.

Rhoni Andres – Assistance with the formatting of this document, creating the posterboard and various visual aids, editing of QB's logo

Sharon Thorpe – Consultant on design of the hand strap to house wearable sensors, fabrication of said hand strap

William Atkinson – 3D printing expert, printed the casing for QB's body for us at cost

Chandler Doe – Freelance graphic designer, Created original design for QB logo

Alex Rodriguez

EDUCATION
Florida Atlantic University, Boca Raton FL
Bachelor of Science in Electrical Engineering Aug 2018
California State University – Sacramento
Bachelor of Science in Electrical Engineering May 2021
GPA: 3.58 / 4.0

TECHNICAL SKILLS
Programming: C/C++, Assembly, Matlab, PSpice
Software: Visual Studio, OrCAD Capture CIS, MS Office, Quartus II, Advanced Design System (ADS), LASI, Code Composer Studio (CCS)
Hardware/Tools: Tektronix Oscilloscope, Digital Multimeter, Signal & Function Generator

ACADEMIC PROJECTS
Digital Lock System *(Intro to Microcontrollers, Fall 2016)*
Developed and assembled a fully functional digital locking system using an MSP430 microcontroller. The application includes digital lock systems and provides a more efficient and safer way to lock and monitor different locations.

- An MSP430 microcontroller was incorporated in the design to control the 4x3 keypad, solenoid and LCD Display to assure that the appropriate code activated the device to lock or unlock when necessary.
- The time frame given to complete the DLS was two weeks but the design was completed within a week. Also, many parts of the original schematics were either not used or unnecessary and therefore made the DLS work more efficiently.
- *Tools used: Visual Studio, Energia, MSP430*

WORK EXPERIENCE
Sears July 2015-January 2017
Consultative Sales Associates

- Assist customers by utilizing approved consultative selling practices and guidelines to identify customer needs and provide appropriate solutions.
- Maintain knowledge of products and use this knowledge to assist and educate customers on options available.
- Provide customer assistance; and deliver a positive customer shopping experience.

MetroPCS April 2014–August 2014 *Retail Sales Associate*

- Analyze customer needs and present value-added solutions while driving profitable sales.
- Provide our customers with a basic understanding of the functionality of the products and accessories they purchase so that they can immediately enjoy their new solutions.
- Support daily business operations, including processing customer transactions and auditing cash receipts daily.

Fluent in Spanish

Victor Lugo

SUMMARY
A Computer Engineering Student with skills in hardware and software who is eager to work in the field. Works well with others and able to take on challenges in a group or independently. Also pursuing a minor in Japanese to widen horizons with technological innovation in Japan as well.

SOFTWARE AND HARDWARE SKILLS
Languages

- Proficient in: C, Verilog, VHDL, Python
- Familiar with: Java, C++, HTML, and Assembly Languages

Software

- Microsoft Visual Studio
- Unix/Linux
- Vivado and Quartus
- DOS(Windows XP)

Experienced with Microcontrollers
Understanding of CPU design through Gates and Circuitry

RELATED SKILLS
Third Year Japanese Student

- Planning to obtain Minor in Japanese

Works Well with Others
Fast Learner

EXPERIENCE IN COURSE WORK
Project: Webserver Server Security System

- Programmed Raspberry Pi with Python and a server framework to connect online. Website programmed with Javascript and HTML. Pi was wired through GPIO to a Parallax Propeller Board which was programmed in C. Both boards were connected to several sensors that updated the website.

Labs for Computer Component Design

- Programming FPGAs w/Verilog & VHDL, DOS for x86 Programs, Cadence for CMOS, etc.

EDUCATION
Los Rios Community College District Fall 2014-Spring 2018
Sacramento, California

- Transfer pursuing BS in Computer Engineering and Minor in Japanese

California State University, Sacramento Fall 2018-Present
Sacramento, California

- Planning to Graduate Spring 2020 with BS in Computer Engineering and Minor in Japanese

Zachary Tapia

Education

California State University, Sacramento- Electrical & Electronics Engineering, GPA: 3.89

Experience

Intel Non-volatile Solutions Group, Client Division Memory, Folsom Ca.- Storage Solutions Architect Intern, Carried out performance testing and analysis of end user memory products.

Senior Electrical & Engineering Student Focused on Analog Circuit analysis and design.

Skills

- C++, Python, Verilog, Assembly/MASM, git/github, AutoIt, Matlab, SPICE
- Electronics lab equipment: scope, FGEN, power supplies etc.
- State machine design with FPGA (STM, Parallax, Raspberry Pi, Intel)
- Prototype breadboarding, Soldering
- Network analysis techniques
- MS Office: Excel, Word, Outlook, PowerPoint
- Oral Presentations, outdoor leadership & education

Career Interests

Seeking opportunities to work with a team learning about and building solutions to problems in microelectronics

Employment History

- **Intel Corporation (NSG, CDM), Solutions Architect Intern**
May 2019 - Present
- **Associated Students Inc, Outdoor Trip Leader**
August 2019 - November 2019
- **Nathan Saucedo Electric.** Apprentice Electrician/Laborer
Oct 2018 - Jan 2019
- **Walgreen Co. Pharmacy Technician**
Feb. 2011 - July 2018

Projects

- Designed scripting software to automate PC benchmarking tasks
- Designed, Built and Tested 2-GHz Patch Antenna matched to 50 Ohm Transmission Line
- Designed Acoustic Tuner Software
- Developed Personal Static Website:
https://razzmazach.github.io/websiteRepo/index.html
- Programmed 4 Way Intersection Traffic Control System

Additional Information

- Licensed HAM Radio Operator Call Sign: KN6CVH
- Lifetime outdoor enthusiast
- Experienced rock climber & mountaineer
- Wilderness First Responder

JAYSON THORPE

Computer Engineer

xxx xxx xxxx

xxxxxxxxxxxx@xxxxx.com

Sacramento, CA

linkedin.com/in/jaysonthorpe

TECHNICAL SKILLS

C, C++, CSS3, Java, JavaScript, Python, HTML5, MySQL, Node

PERSONAL SKILLS

WRITTEN & VERBAL COMMUNICATION

ATTENTION TO DETAIL

ADAPTABLE & RELIABLE

PROBLEM SOLVER

ABILITY TO GRASP NEW CONCEPTS QUICKLY

SELF MOTIVATED

CUSTOMER FOCUSED

PROFILE & OBJECTIVE

Highly motivated computer engineering graduate seeking a position with a company that can benefit from my programming experience, customer service skills, and desire to learn. I thrive in solution-driven environments and thoroughly enjoy working in both team and solo settings to accomplish goals.

EDUCATION

BS, COMPUTER ENGINEERING — 2018 - 2021

CALIFORNIA STATE UNIVERSITY, SACRAMENTO

Relevant Coursework: Data Structures and Algorithm Analysis, Network Analysis, Systems Programming in UNIX, Software Engineering, OS Principles

3.5 GPA | Dean's List

RELEVANT PROJECTS

DESIGN : THE QUARANTINE BUDDY (QB) — 04/2021

CPE 190/191 : SENIOR DESIGN PROJECT 1 + 2

Designed and built a virtual companion focused on stress relief for those having a difficult time in isolation during the pandemic. Based on readings from a wearable device with embedded sensors, QB will interact with the user when they are stressed to suggest and assist in several ways for the user to reduce and manage stress levels. Alexa Voice Services functionality was incorporated to facilitate the user voice interface. QB's sensor array is made up of sensors for temperature, heart rate, and a galvanic skin response to take user biometric readings. These take their respective measurements at regular intervals which are then handled and processed by an Arduino Lilypad to generate a user Stress Score. If the Stress Score exceeds a given threshold, a Moment of Stress (MoS) is signaled. The information generated by the sensor Arduino is then wirelessly transmitted via Bluetooth to the main body of QB.

USER APPLICATION: OSCAR VISION — 12/2020

CSC 131 : COMPUTER SOFTWARE ENGINEERING

Application for user to search Oscar-nominated films, including filter capability, search engine, and GUI. Wrote the search engine in Java for this project, which enforced Scrum principles and job roles within a group development setting. Project included concepts and techniques for specifying requirements, architectural design specifications, prototyping, documentation, and integration to a final product.

OPERATING SYSTEM: KUDOS — 05/2021

CPE 159 : OPERATING SYSTEM PRAGMATICS

Using operating system principles, designed and implemented a multi-tasking operating system called kudOS. Project included the scheduling of processes, control and allocation of computer resources, and user interfacing.

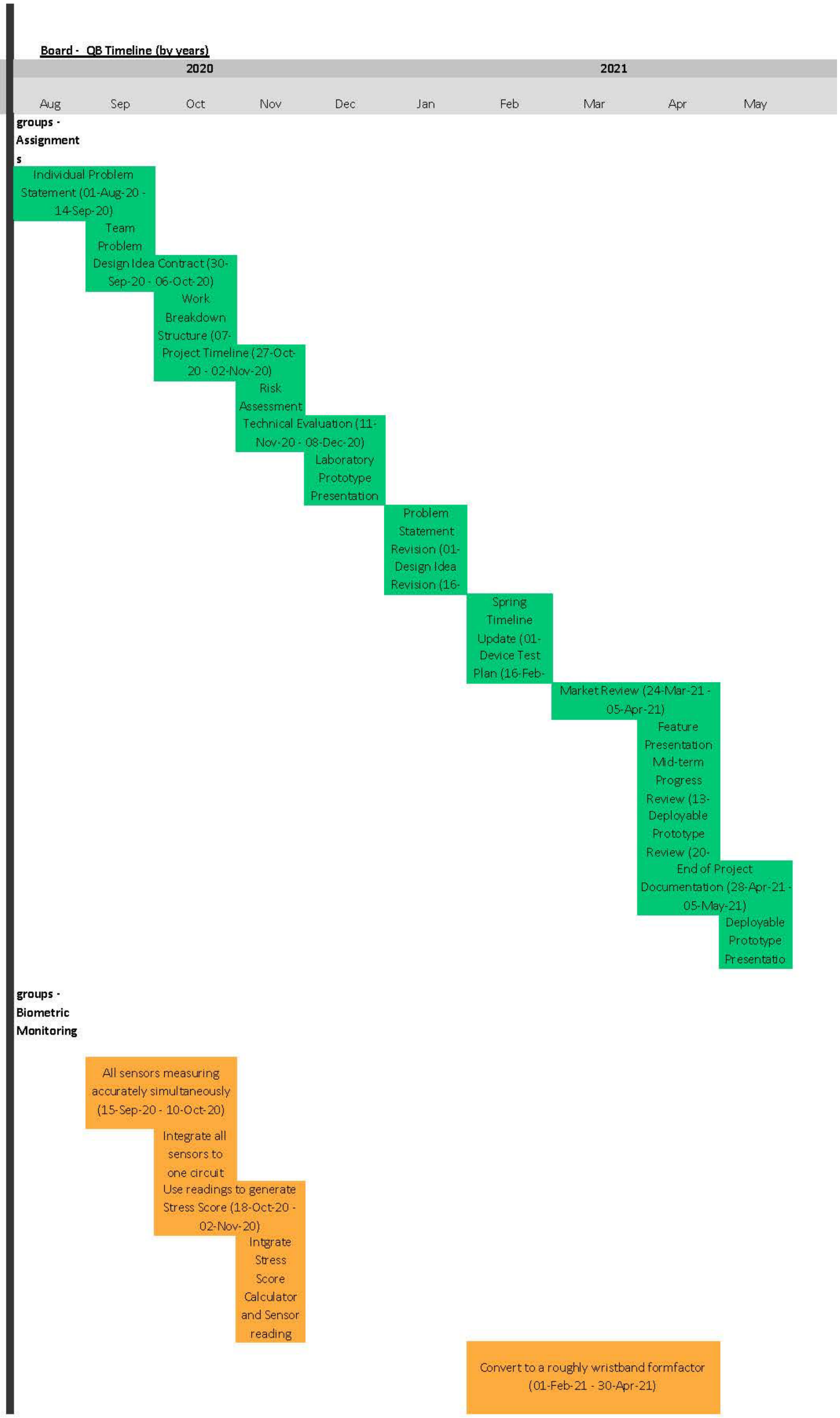
RECENT WORK HISTORY

CARVANA | *SACRAMENTO, CA* — 09/2019 - present

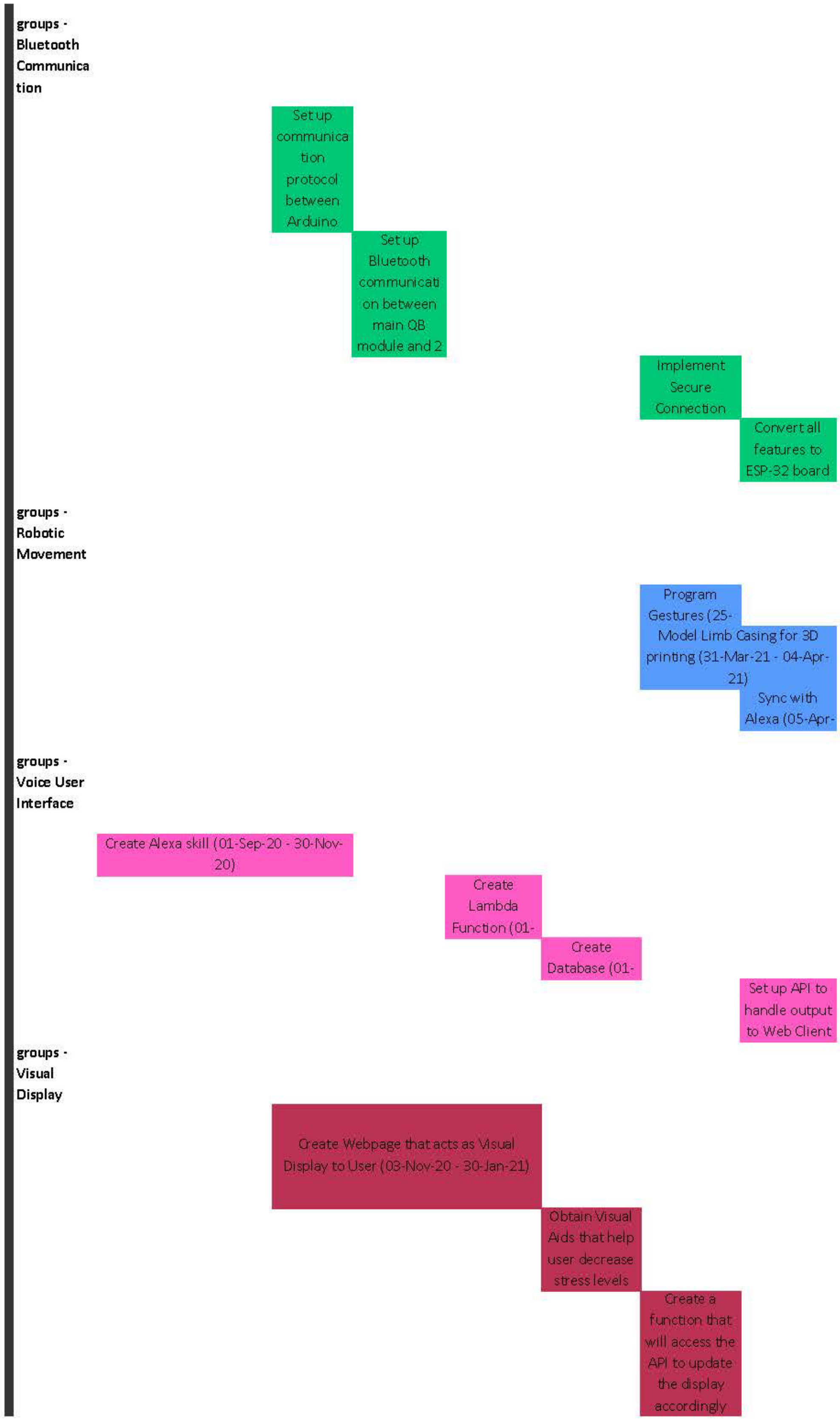
CUSTOMER ADVOCATE

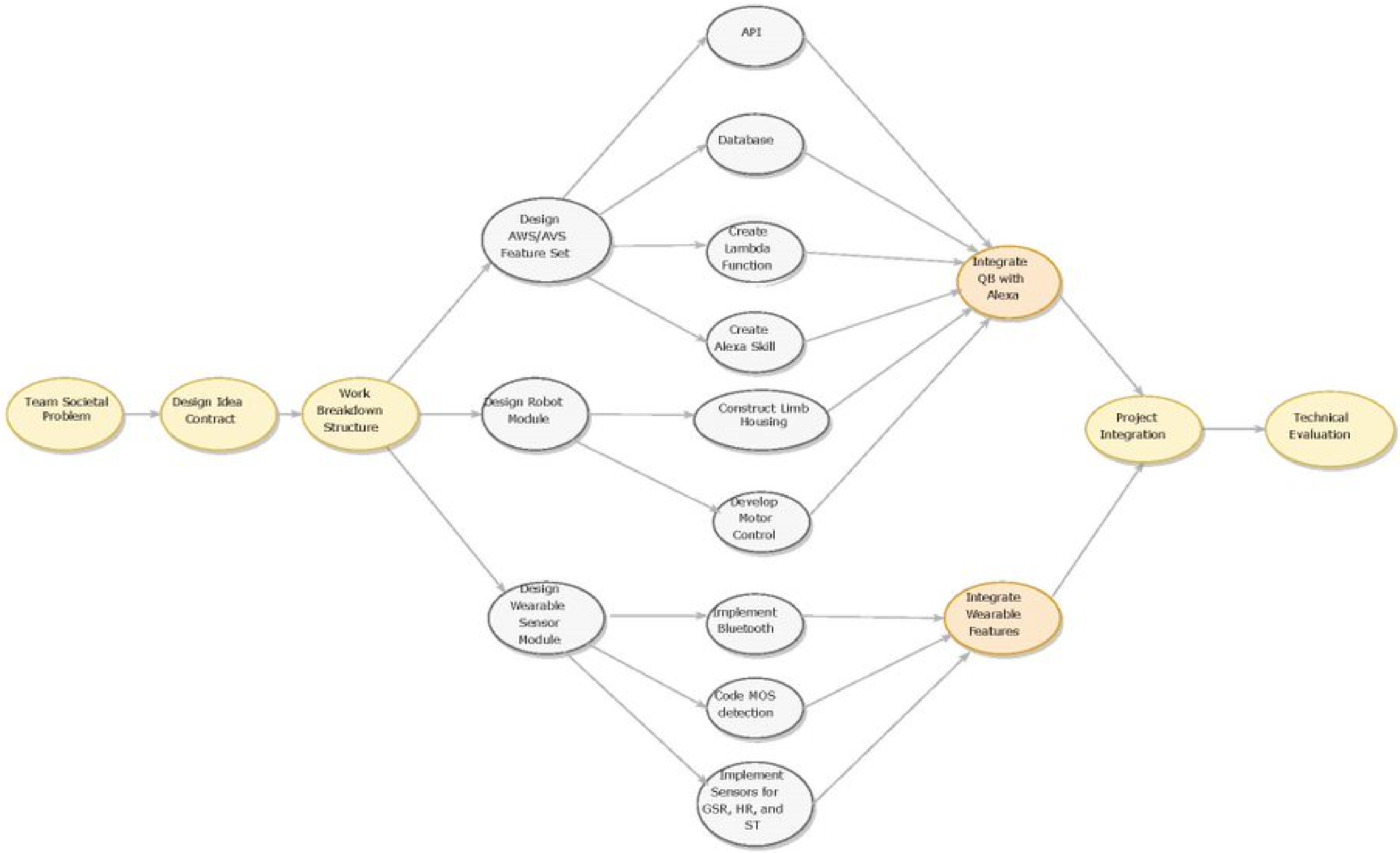
Appendix G. Project Timeline & Work Breakdown Structure

	Start	End
groups - Assignments		
Individual Problem Statement	01-Aug-2020	14-Sep-2020
Team Problem Statement	15-Sep-2020	29-Sep-2020
Design Idea Contract	30-Sep-2020	06-Oct-2020
Work Breakdown Structure	07-Oct-2020	26-Oct-2020
Project Timeline	27-Oct-2020	02-Nov-2020
Risk Assessment	03-Nov-2020	10-Nov-2020
Technical Evaluation	11-Nov-2020	08-Dec-2020
Laboratory Prototype Presentation	09-Dec-2020	12-Dec-2020
Problem Statement Revision	01-Jan-2021	15-Jan-2021
Design Idea Revision	16-Jan-2021	30-Jan-2021
Spring Timeline Update	01-Feb-2021	15-Feb-2021
Device Test Plan	16-Feb-2021	23-Feb-2021
Market Review	24-Mar-2021	05-Apr-2021
Feature Presentation	06-Apr-2021	12-Apr-2021
Mid-term Progress Review	13-Apr-2021	19-Apr-2021
Deployable Prototype Review	20-Apr-2021	27-Apr-2021
End of Project Documentation	28-Apr-2021	05-May-2021
Deployable Prototype Presentation	06-May-2021	20-May-2021
groups - Biometric Monitoring		
All sensors measuring accurately simultaneously	15-Sep-2020	10-Oct-2020
Integrate all sensors to one circuit	11-Oct-2020	17-Oct-2020
Use readings to generate Stress Score	18-Oct-2020	02-Nov-2020
Intgrate Stress Score Calculator and Sensor reading into one module	02-Nov-2020	22-Nov-2020
Convert to a roughly wristband formfactor	01-Feb-2021	30-Apr-2021



groups - Bluetooth Communication		
Set up communication protocol between Arduino Modules	15-Nov-2020	30-Nov-2020
Set up Bluetooth communication between main QB module and 2 device	01-Dec-2020	06-Dec-2020
Implement Secure Connection	01-Mar-2021	21-Mar-2021
Convert all features to ESP-32 board	01-Apr-2021	30-Apr-2021
groups - Robotic Movement		
Program Gestures	25-Mar-2021	30-Mar-2021
Model Limb Casing for 3D printing	31-Mar-2021	04-Apr-2021
Sync with Alexa	05-Apr-2021	11-Apr-2021
groups - Voice User Interface		
Create Alexa skill	01-Sep-2020	30-Nov-2020
Create Lambda Function	01-Jan-2021	31-Jan-2021
Create Database	01-Feb-2021	28-Feb-2021
Set up API to handle output to Web Client	01-Apr-2021	30-Apr-2021
groups - Visual Display		
Create Webpage that acts as Visual Display to User	03-Nov-2020	30-Jan-2021
Obtain Visual Aids that help user decrease stress levels	01-Feb-2021	27-Feb-2021
Create a function that will access the API to update the display accordingly	01-Mar-2021	27-Mar-2021




Team Societal Problem
Design Idea Contract
Work Breakdown Structure
Design AWS/AVS Feature Set
Design Robot Module
Design Wearable Sensor Module
API
Database
Create Lambda Function
Create Alexa Skill
Construct Limb Housing
Develop Motor Control
Implement Bluetooth
Code MOS detection
Implement Sensors for GSR, HR, and ST
Integrate QB with Alexa
Integrate Wearable Features
Project Integration
Technical Evaluation

PERT CHART

Level 1	Level 2	Level 3	Level 4
0. Team Activity Reports			
	0.1: Individual Reports		
		0.1.1: Compile tasks/hours for week	
		0.1.2: Project tasks/hours for next week	
	0.2: Team Report		
		0.2.1: Compile individual reports	
		0.2.2: Compile group meeting topics/ minutes	
1A: Individual Problem Statement			
	1A.1: Write Individual Report		
		1A.1.1: Research of Problem	
1B: Team Problem Statement			
	1B.1: Write Group Report		
		1B.1.1: More in-depth research of Problem	
		1B.1.2: Develop Design Concept	
	1B.2: Format Report		
2: Design Idea Contract			
	2.1: Finalize Design Idea		
		2.1.1: Finalize Feature Set	
			2.1.1.1: Research methods of measuring stress
			2.1.1.2: Research methods of

			treating/alleviating stress
			2.1.1.3: Research methods of detecting stress
		2.1.2: Research/Purchase Components	
	2.2: Write Report		
		2.2.1: Research Different Approaches	
	2.2: Format Report		
	2.3: Give Presentation on Problem		
		2.3.1: Create PowerPoint presentation	
		2.3.2: Write notes for verbal presentation	
3: Work Breakdown Structure			
	3.1: Create Work Breakdown Chart		
		3.1.1: Breakdown all Tasks to smaller subtasks	
	3.2: Write report		
	3.3: Format Report		
4. Project Timeline			
	4.1: Gantt Chart		
		4.1.1: Group tasks into families	
		4.1.2: Estimate timeline	
		4.1.3: Assign Team Members	
	4.2: PERT Diagram		
		4.2.1: Establish project dependencies	
			4.2.1.1: Project Milestones
	4.3: Write Report		
	4.4: Format Report		
5. Risk Assessment			

	5.1: Assess Risks		
		5.1.1: Identify critical paths	
		5.1.2: identify potential events/risks	
		5.1.3: List possible mitigation strategies	
		5.1.4: Risk assessment chart	
		5.1.5: Examine effect of social distancing on project	
	5.2: Write report		
	5.3: Format Report		
6. Technical Evaluation			
	6.1: Presentation		
		6.1.1: Prepare technical discussion	
		6.1.2: Compile stats for semester	
		6.1.3: Test and confirm working feature set	
7. Laboratory Prototype Presentation			
	7.1: Presentation		
		7.1.1: Write presentation outline	
		7.1.2: Prepare visual aid	
		7.1.3: Test and confirm working features for demo	
8A. Problem Statement Revision			
	8A.1: Revise Problem Statement		
		8A.1.1: Reexamine the problem	
		8A.1.2: Adjust accordingly	
	8A.2: Write Report		
	8A.3: Format Report		

8B. Design Idea Revision			
	8B.1: Revise Design Idea		
		8B.1.1: Reexamine design idea against revised problem statement	
	8B.2: Write Report		
	8B.3: Format Report		
8C. Spring Timeline Update			
	8C.1: Revise Timeline		
		8C.1.1: Evaluate status of project	
		8C.2.2: Adjust accordingly	
	8C.2: Presentation		
		8.4.1: Prepare presentation outline	
		8.4.2: Prepare presentation PowerPoint	
	8C.3: Write Report		
	8C.4: Format Report		
9. Device Test Plan			
	9.1: Establish Appropriate Test Conditions		
		9.1.1: Plan test for Sensors	
		9.1.2: Plan test for wireless transmission	
		9.1.3: Plan test for robotic movement	
		9.1.4: Plan test for user voice interface	
	9.2: Write report		
	9.3: Format report		
10. Market Review			
	10.1: Research Marketability Forecast		
	10.2: Write Report		

	10.3: Format Report		
	10.4: Presentation		
		10.4.1: Write presentation outline	
		10.4.2: Prepare presentation PowerPoint	
11. Feature Presentation			
	11.1: Present Features		
		11.1.1: Confirm working features for presentation	
		11.1.2: Write presentation Outline	
		11.1.3: Prepare presentation PowerPoint	
	11.2: Write Report		
	11.3: Format Report		
12. Mid-Term Progress Review			
	12.1: Revise test plan if needed		
	12.2: Write Report		
	12.3: Format Report		
13. Deployable Prototype Review			
	13.1: Post-Project Audit		
	13.2: Prototype Demonstration		
		13.2.1: Prepare project for demonstration	
		13.2.2: Write presentation outline	
		13.2.3: Prepare presentation visual aid	
14. End of Project Documentation			
	14.1: Write Final Report		

	14.2: Format Final Report		
15. Deployable Prototype Presentation			
	15.1: Make Poster		
	15.2: Ensure readiness of prototype		
	15.3: Demo Prototype		
16. Biometric Monitoring			
	16.1: All sensors measuring accurately simultaneously		
		16.1.1: HR sensor takes BPM	
			16.1.1.1: Code, test, debug
		16.1.2: HR sensor takes HRV	
			16.1.2.1: Code, test, debug
		16.1.3: Temp Sensor	
			16.1.3.1: Code, test, debug
		16.1.4: GSR Sensor	
			16.1.4.1: Code, test, debug
		16.1.5: Integrate all sensors to one circuit	
			16.1.5.1: Code, test, debug
	16.2: Use readings to generate Stress Score		
		16.2.1: Research algorithms using sensor data for stress scores	
		16.2.2: Flowchart algorithm	
			16.2.2.1: Chart 5 rules for weighing aspects of GSR and ST

		16.2.3: Research HR as additional metric	
			16.2.3.1: integrate into MOS flowchart
		16.2.4: Translate flowchart into Arduino C++ code	
			16.2.4.1: Write Code
			16.2.4.2: Debug
			16.2.4.3: Test and debug with artificial test values
			16.2.4.4: Test with actual user
17. Bluetooth Wireless Communication			
	17.1: Set up communication protocol between Arduino modules		
		17.1.1: Use correct AT commands	
		17.1.2: Establish Master/Slave	
		17.1.3: Pair and communicate data serially	
	17.2: Set up Bluetooth communication between main QB module and 2nd development kit (ESP32 or Maxrefdes100)		
		17.2.1: For ESP32, program for Arduino in similar manner	
		17.2.2: If MAXREF use C and mbed programming environment	
18. Robotic Movement			

	18.1: Program Gestures		
		18.1.1: Greeting	
		18.1.2: Nod	
		18.1.3: Fist Bump	
	18.2: Model Limb Casing for 3D printing		
	18.3: Synch w/ Alexa		
19. Voice User Interface			
	19.1: Create Alexa Skill		
		19.1.1: Create Invocation Name	
		19.1.2: Create Intents for each interaction	
			19.1.1.1: Create Dialog Confirmation
		19.1.3: Create Slots that manage input from the user	
			19.1.3.1: Create Slot Filling/Confirmation
			19.1.3.2: Create Slot Validation Rule
	19.2 Create Lambda function		
		19.2.1: Format Lambda function into Node.js language	
		19.2.2: Write the handlers/functions/sessions that will serve as output to the user.	
		19.2.3: Create/Validate lambda function responsible for reading and writing from/to a database	
			19.2.1.1: Link Alexa Skill Kit as a trigger to the main lambda function

			19.2.3.1: Create a separate Lambda function that controls reading from the database
	19.3: Create Database		
		19.3.1: Create a DynamoDB table	
			19.3.1.1: Set up permissions to access write/read access via Lambda Function
		19.3.2: Set up attributes and data value types	
	19.4: Set up API to handle output to web client		
		19.4.1: Create an API using API Gateway Console in AWS	
		19.4.2: Link the API to the Lambda function that will read values from the database	
		19.4.3: Deploying the API	
			19.4.3.1: Enabling Cross-Origin Resource Sharing
20. Visual Display			
	20.1: Creating Website that will work as Visual Display for User		
		20.1.1: Use codepen.io to create website	
			20.1.1.1: Write JS, CSS and HTML code for website
	20.2: Obtain Visual Aids that will help		

	user decrease stress levels		
		20.2.1: Relaxing Music	
		20.2.2: Breathing Exercises	
		20.2.3: General Body exercises	
	20.3: Create a function that will access the API to update the Display accordingly		